Linear regression and learning: contributions to regularization and aggregation methods

Raphaël Deswarte

université PARIS-SACLAY

ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

NNT : 2018SACLX047

Thèse de doctorat

Régression linéaire et apprentissage : contributions aux méthodes de régularisation et d'agrégation

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'École polytechnique

École doctorale n°574 École Doctorale de Mathématiques Hadamard (EDMH)
Laboratoire : Centre de Mathématiques Appliquées (CMAP)

Spécialité de doctorat : Mathématiques appliquées

Thèse présentée et soutenue à Palaiseau, le 27 septembre 2018, par

RAPHAËL DESWARTE

Composition du Jury :

Pierre Alquier Professeur des Universités, ENSAE	Président
Olivier Wintenberger Professeur des Universités, Sorbonne Université	Rapporteur
Vincent Rivoirard Professeur des Universités, Université Paris-Dauphine	Rapporteur
Véronique Gervais-Couplet Ingénieure de recherche, IFP Énergies Nouvelles	Examinatrice
Tim van Erven Professeur assistant, Universiteit Leiden	Examinateur
Karim Lounici Professeur des Universités, École polytechnique	Examinateur
Guillaume Lecué Professeur des Universités, ENSAE	Co-directeur de thèse
Gilles Stoltz Directeur de recherche, CNRS – Université Paris-Sud	Co-directeur de thèse

Remerciements

En plus d'être une aventure scientifique, le doctorat est une aventure humaine, et même une tranche de vie. Mon parcours de doctorant m'a permis de côtoyer de nombreuses belles personnes, que je souhaite saluer et remercier ici.

Je me limiterai aux personnes rencontrées dans le cadre de mon doctorat ou des activités qui y sont liées. Les proches que je connais depuis longtemps, ou que j'ai rencontrés par ailleurs, ne sont donc pas cités ici, je pense néanmoins bien à eux.

Ma gratitude va d'abord à mes encadrants, Guillaume Lecué et Gilles Stoltz.
Vous avez su réaliser un encadrement complet : à la fois de grande qualité scientifique, adapté à ma situation, m'aidant à construire un projet professionnel, et enfin, très humain. Vous avez été des rayons de soleil dans des périodes parfois difficiles.

Guillaume, tu as toujours été ouvert à mes idées et encourageant pour les mettre en œuvre. Ton optimisme et ton enthousiasme m'ont motivé tout au long de ce doctorat.

Gilles, ton pragmatisme rassurant et tes conseils précieux m'ont aiguillé efficacement et m'ont amené à découvrir d'autres manières de réfléchir. Merci également pour ta relecture très minutieuse de mes documents.

Pour avoir aidé nombre de doctorants ne bénéficiant pas d'un encadrement de cette qualité, je mesure ma chance d'avoir eu deux encadrants tels que vous. Merci encore.

J'ai également de la gratitude envers mes "grands frères de thèse" Sébastien, Pierre, Paul et Emilien : vos conseils et vos encouragements m'ont bien aidé ! J'inclus également dans ce salut amical mes "petits frères et sœur de thèse", notamment Pierre, Malo, Margaux : bonne fin de doctorat !

Je remercie sincèrement les membres de mon jury.

Vincent, Olivier, merci d'avoir pris le temps de relire en détails ma thèse et d'en rédiger les rapports.

Véronique, notre collaboration appliquée a été fort intéressante ; j'espère qu'elle portera de nombreux fruits à court et long termes. Que Sébastien et Charles-Pierre, qui avaient posé les prémices de cette collaboration, soient ici remerciés également.

Karim, je vous remercie d'avoir accepté de faire partie du jury de ma soutenance.

Pierre, vous avez été l'enseignant qui m'a ouvert les portes de l'apprentissage statistique, c'est donc un plaisir pour moi de vous compter dans mon jury.

Tim, ik heb veel aan je te danken. Je bent een ontzettend betrokken begeleider geweest tijdens mijn verblijf, altijd met een glimlach en gemotiveerd. Dankzij jou heb ik één mooi hoofdstuk meer in mijn thesis en heb ik een prachtige tijd gehad in Nederland. Mijn verblijf in Leiden was één van de mooiste tijden, niet alleen van mijn PhD, maar ook van mijn hele leven. Ik heb een erg plezierig land ontdekt, met geweldige mensen.

Dirk, het was een plezier om met je samen te werken, ik zal je enthousiasme niet snel vergeten. Een van je favoriete bijvoeglijke naamwoorden "Awesome", is duidelijk op jou van toepassing! Het was een groot genot om ook de andere PhD studenten te ontmoeten: Anja, Niels, Sanne, Stephanie, Yuki, Julian...

Ook heb ik zeer fijne uren doorgebracht bij de danslessen van de sportschool, blij om Ymkje, Fleur, Jonathan en Robin te hebben leren kennen.

Leiden is zonder twijfel een fantastische stad met geweldige mensen!

Au sein du CMAP, je remercie Nasséra et Alex pour leur support administratif bien utile. Merci également à Anne et Thierry pour leur suivi attentif de ma situation.

Je salue les doctorants qui m'ont accompagné durant ces quelques années : Tristan, Gaoyue, Andrea, Jaouad, Etienne, Hélène, Manon, Aymeric, Romain, Massil, Raphaël, Jean-Bernard, Perle, Rémi...

Ayant eu le plaisir d'enseigner à Polytechnique (et aussi dans le Master 2 MVA), j'aimerais saluer à la fois les autres enseignants avec qui j'ai collaboré (particulièrement Alexandre Tsybakov et Eric Moulines), et mes étudiants : Francis, Laure, Marie, Etienne, Lise, Gabriel...

Ma période de doctorat a été aussi pour moi l'occasion de m'investir au service des étudiants, sous plusieurs formes.

J'ai eu l'honneur de siéger au Conseil d'Administration de l'Ecole polytechnique. Je tiens à saluer respectueusement le Président Jacques Biot, et les Directeurs Généraux Yves Demay et François Bouchet, ainsi que les membres successifs du Conseil, avec une pensée particulière pour certains avec qui j'ai beaucoup échangé : Aldjia, Sylvie, Emmanuel, Jean-Baptiste, Sylvain, Omar, Sébastien.

Je salue également tous les gens avec qui j'ai collaboré au service des étudiants : "Kès" successives (notamment la "Dolkès Vita" et la "Kèsdorado"), "Graduate School", pôle Diversité et Réussite, service des sports... Je souhaite le meilleur à cette école merveilleuse qu'est Polytechnique et à toute sa communauté.

Du côté des doctorants, je salue amicalement mes compères de l'association X'Doc, désormais Doc'Union : Antoine, Nicolas, Joris, Loann... en pensant à tous les événements que nous avons organisés ensemble.

J'ai découvert au cours de ma thèse la Confédération des Jeunes Chercheurs (CJC). La compétence et l'investissement discret mais sans faille de ses membres (Clément, Aurélien,...) ont été pour moi de vraies leçons d'humilité et d'altruisme.

J'ai une pensée pour mes co-listiers, représentants des doctorants au Conseil Académique de Paris-Saclay : Mathilde, Jean-François, Thomas, Valérie... Ce fut un plaisir d'agir à vos côtés en faveur des doctorants. Plus généralement, je salue les membres du Conseil Académique, qui n'ont pas ménagé leurs efforts pour construire Paris-Saclay. Puisse cette belle idée de collaboration dans Paris-Saclay aboutir à des vraies avancées concrètes améliorant le quotidien et les conditions de travail de toute sa communauté : étudiants, personnel...

La fin de la rédaction de ma thèse s'étant effectuée en parallèle d'un travail de data scientist, j'en profite pour saluer mes collègues de l'équipe INS/DAT : Jean, Clément, Eric, Khémon, Pierre, Charlotte, Laurent, Guillaume, Lucile, Chloé, Jaouad, Lorraine, Dimitri... ainsi bien sûr que le chef de l'équipe, Louis.

Je souhaiterais rendre hommage à quelques-uns de mes enseignants en mathématiques, qui ont marqué mon parcours chacun à leur manière : Mme Orgogozo (qui sait mettre en évidence le plaisir des mathématiques), Mme Lasserre (d'une rigueur exemplaire), Mme Gouteyron (dont certaines discussions m'ont amené définitivement à choisir les mathématiques), Mme Picard et M. Alquier (qui m'ont initié aux statistiques et à l'apprentissage).

Enfin, je conclurai par une pensée très tendre et affectueuse pour mes parents. Cette thèse vous est dédiée.

Have a nice reading! Veel leesplezier! Bonne lecture !

Table des Matières

Chapter 1

Introduction générale

1.1. Présentation générale de la thèse

Cette thèse se situe dans le domaine des statistiques mathématiques, avec une focalisation particulière sur l'apprentissage. Son fil conducteur est la régression linéaire, dont elle cherche à mettre en valeurs la multiplicité des approches et des variations selon le problème statistique à traiter —mais aussi les logiques communes. Elle s'intéresse en particulier aux bénéfices de relier les cadres (notamment "batch stochastique" et "séquentiel déterministe") entre eux et d'en transférer des idées.

Deux types de méthodes jouent un rôle central dans les travaux présentés : les approches dites "par régularisation", et celles dites "par agrégation".

Cette introduction générale (chapitre 1) présente les différents cadres dans lesquels se situent les travaux de recherche et leurs enjeux. Elle indique également les apports de cette thèse, et met en évidence des pistes de recherche possibles pour le futur.

L'introduction mathématique (chapitre 2) présente les outils mathématiques utilisés dans les différents chapitres.

Le chapitre 3 présente une construction d'une régularisation optimale dans un cadre dit "batch stochastique", améliorant notamment, selon certains critères qui seront présentés, l'estimateur LASSO, très utilisé à l'heure actuelle.

Le chapitre 4 s'intéresse à un cadre plus général, en un certain sens, que la régression linéaire : l'optimisation convexe séquentielle. Nous présentons tout d'abord des améliorations pour un algorithme récent et prometteur, MetaGrad, et ensuite proposons une approche pour convertir cet algorithme d'un des cadres de cette thèse ("séquentiel", utilisé aux chapitres 5 et 6) vers l'autre cadre ("batch", utilisé au chapitre 3).

Le chapitre 5 aborde un problème a priori classique, la création d'intervalles de prévision, mais dans un cadre original : les suites individuelles. Il propose une méthodologie nouvelle, permettant d'adapter les algorithmes existants. Cette méthodologie requiert une optimisation, dont on présente des méthodes pour l'effectuer sur trois algorithmes (régression Ridge, algorithme EWA et son extension Fixed-Share EWA).

Le chapitre 6 met en œuvre des algorithmes d'agrégation en suites individuelles, et de prévision par intervalles (selon la méthodologie développée au chapitre 5), sur un jeu de données de production pétrolière.

Les différents chapitres ont été rédigés de façon à pouvoir être lus indépendamment.

1.2. Cadre mathématique de la thèse

1.2.1. Différents cadres pour la régression linéaire, adaptés à différentes situations

Régression linéaire

La régression linéaire est une modélisation d'une grandeur y (souvent appelée "variable à expliquer") par une combinaison linéaire $\widehat{y}$ de grandeurs connues $x_1, \dots, x_d$ (dites "variables explicatives") :

$$\widehat{y} = \sum_{i=1}^{d} w_i x_i.$$

Cela peut permettre de mieux comprendre la grandeur étudiée y, mais également d'effectuer des prévisions.

Le travail du statisticien/de la statisticienne consiste à déterminer judicieusement le vecteur des coefficients : $(w_1, \dots, w_d) \in \mathbb{R}^d$.

C'est une des méthodes statistiques les plus anciennes : Stanton [2001] indique que les premiers germes remontent à [Galton, 1894], et [Pearson, 1896] ; et d'après Stigler [1986] certaines idées liées, telles que la notion de moindres carrés, ont été présentées encore antérieurement par Gauss et Legendre. Il s'agit aussi de l'une des approches les plus utilisées dans tous les domaines des sciences.

Tous les éléments évoqués : processus de prévision, modélisation des variable à expliquer, et variables explicatives, influent sur la méthodologie à mettre en œuvre. Les sections suivantes présentent leur impact dans les différents chapitres de la thèse, qui font chacun intervenir plusieurs cadres. Les figures 1.1 et 1.2 résument le positionnement des divers chapitres.

Différents processus de prévisions

Plusieurs situations existent quant aux processus de prévision eux-mêmes. Deux grands cas peuvent se présenter.

Cadre "batch". Dans le cadre dit "batch" (Protocole 1), le but est de réaliser une prévision unique (ou éventuellement une série de prévisions d'une seule traite), à partir d'un échantillon d'apprentissage (imposé ou choisi). Il n'y a pas de retour d'expérience intermédiaire, on a dès le début la totalité (le "lot" –"batch" en anglais, d'où le nom) des informations aidant à prévoir. Le chapitre 3 s'inscrit dans ce cadre avec des hypothèses stochastiques sur les données ; le chapitre 5, lui, s'attaque à la question d'une série de prévisions "batch" sans ce type d'hypothèses.

Cadre "séquentiel". L'autre cas correspond à un processus de prévision dit "séquentiel" ("online" en anglais, c'est-à-dire "en ligne"), décrit dans le Protocole 2. Il s'agit d'effectuer plusieurs prévisions, en disposant d'un retour d'expérience entre deux prévisions successives, typiquement la "valeur réellement observée" de la variable que l'on avait tenté de prévoir précédemment (ou par exemple, dans le chapitre 4, le gradient de la perte). On dispose parfois

Protocole 1 Processus de prévision "batch"

Phase d'apprentissage :

Le statisticien dispose d'un échantillon d'apprentissage $(x_1, y_1), \ldots, (x_N, y_N)$

Phase de prévision :

1. L'environnement révèle le vecteur x des variables explicatives
2. Le statisticien propose une valeur $\widehat{y}$ en utilisant l'échantillon d'apprentissage $(x_1, y_1), \ldots, (x_N, y_N)$
3. L'environnement révèle la vraie valeur y
4. Le statisticien subit la perte $\ell(\widehat{y}, y)$

d'un échantillon d'apprentissage avant la première prévision –mais pas toujours. L'objectif est bien entendu d'essayer de prévoir correctement à chaque tour, mais la performance sera surtout évaluée sur l'erreur cumulée, plutôt que sur une prévision précise. On s'intéresse ainsi à la perte cumulée :

$$L_T(\widehat{y}_1, \ldots, \widehat{y}_T) := \sum_{t=1}^{T} \ell_t(\widehat{y}_t)$$

où les ℓ_t sont des fonctions de perte, correspondant souvent à l'erreur : $\ell_t(\widehat{y}_t) = \ell(\widehat{y}_t, y_t)$, parfois à un coût.

De nombreuses situations réelles peuvent être vues dans ce cadre : prévisions heure par heure (consommation nationale d'électricité : Devaine et al. [2013]), quotidiennes (prévision de propagation de maladie : Chan et al. [2015]), mensuelles, trimestrielles (indicateurs économiques tels que la croissance : Bessec [2010] s'intéresse à des données mensuelles et trimestrielles), annuelles (prévision des coûts liés à une infrastructure : Vernet [2015])...

Dans le cadre de cette thèse, on s'intéresse dans le chapitre 6 au problème de la prévision mensuelle de plusieurs grandeurs liées à la production de pétrole dans un champ pétrolifère, sur une période de dix ans.

Influence de la modélisation, ou non, des données

Modélisation stochastique. La connaissance que l'on possède sur les phénomènes étudiés et la façon dont les données (notamment les variables à prévoir) sont générées permet parfois de proposer une modélisation stochastique, ce qui permet d'obtenir des résultats assez précis "avec grande probabilité" ou en espérance (en moyenne). Ainsi, au chapitre 3, on dispose d'observations Y_i et de vecteurs explicatifs X_i dont on sait qu'il existe un vecteur t^* tel que :

$$Y_i = \langle t^*, X_i \rangle + \xi_i$$

où $\xi_i \sim \sigma\mathcal{N}(0, 1)$ avec σ connu, mais où t^* est inconnu et cherché. On présente alors une méthodologie "sur-mesure" et optimale (en un sens qui sera défini) permettant de tirer parti de cette "bonne" connaissance des données.

Protocole 2 Processus de prévision "séquentiel" (ou "online")

Phase d'apprentissage facultative :

Le statisticien dispose parfois d'un échantillon d'apprentissage.

Phase de prévision :

L'erreur cumulée initiale est nulle : $L_0 = 0$.

for $t = 1, 2, \ldots$

1. L'environnement révèle le vecteur des variables explicatives x_t
2. Le statisticien propose une valeur $\widehat{y}_t$ en utilisant les valeurs observées précédemment $y_1, \ldots, y_{t-1}$ et les variables explicatives présentes et passées $x_1, \ldots, x_t$ (ainsi que l'échantillon d'apprentissage s'il en dispose)
3. L'environnement révèle la vraie valeur y_t
4. Le statisticien subit la perte $\ell(\widehat{y}_t, y_t)$, sa perte cumulée est incrémentée : $L_t = L_{t-1} + \ell(\widehat{y}_t, y_t)$

Suites individuelles. En revanche, dans certaines situations, on ne fait pas de modélisation stochastique sur les données, que l'on considère alors comme des suites déterministes. Cela peut venir d'une connaissance insuffisante des phénomènes étudiés, d'une volonté de "limiter les risques", ou permettre de gérer des situations antagonistes, où la valeur de la variable observée dépend (de manière défavorable) de la prévision qui en a été faite.

Les hypothèses peuvent par exemple se limiter à des bornes sur les pertes, de type :

$$\forall\, t = 1..T, \quad \forall\, y \in \mathbb{R}, \quad \ell_t(y) \in [m, M].$$

On parle pour ce type de modèles de suites individuelles (ou suites arbitraires), c'est le cadre des chapitres 5, 6 et d'une partie du chapitre 4 (ce dernier traitant d'un cadre un peu plus général, l'optimisation convexe séquentielle). Dans ce domaine, la monographie Cesa-Bianchi and Lugosi [2006] constitue une référence importante.

Adaptation pratique de l'approche "suites individuelles". Il faut noter que cette approche, "pire cas" dans l'état d'esprit, est par définition très conservative. Dans sa mise en œuvre pratique sur des jeux de données réels, quelques adaptations plus souples et "optimistes", permettant de mieux s'adapter au degré de stochasticité des données, aboutissent généralement à un gain de performance. En effet, hors situations antagonistes évoquées plus haut, dans le monde physique "le pire n'est jamais certain" et n'est pas le résultat le plus fréquent...

Typiquement, on choisira parfois les paramètres des algorithmes en fonction des données observées plutôt qu'en fonction des bornes théoriques, par exemple en choisissant pour une prévision à un instant t, une calibration de paramètre qui aurait donné de bons résultats jusqu'à l'instant précédent $t - 1$.

Comparaison des cadres "batch stochastique" et "suites individuelles"

Dans les deux cas on a tendance à chercher à se rapprocher des vecteurs "performants", c'est pourquoi on retrouve nombre d'algorithmes similaires. Le lien est même plus profond, avec des résultats théoriques transposables d'un cadre à l'autre (voir les sections de cette thèse évoquant la conversion "online-to-batch"). Mais le raisonnement théorique sous-jacent diffère quelque peu. Dans le cadre batch stochastique, la performance sur la période d'apprentissage donne des indications sur la loi des performances. Dans le cadre des suites individuelles, "les performances passées ne présagent pas des performances futures", mais faire des prévisions proches des experts performants jusqu'alors permet d'éviter des différences fortes avec eux et donc d'empêcher le regret (voir Section 1.2.2) d'augmenter fortement.

	Batch	**Séquentiel**
Modélisation stochastique des données	Chapitre 3 Section "Online to Batch" 4.3 du chapitre 4	Section "Online to Batch" 4.3 du chapitre 4
Modélisation déterministe des données	Chapitre 5 Section "Intervalles de prévision" 6.5 du chapitre 6	Chapitre 4 Chapitre 5 Chapitre 6

Figure 1.1: Différents cadres pour les processus de prévision et la modélisation des données

Influence des variables explicatives

Les connaissances sur les variables explicatives jouent un rôle déterminant : les a priori que l'on a sur leurs capacités prédictives, notamment la confiance que l'on porte (ou non) à leurs performances individuelles, peuvent amener à limiter la recherche des coefficients de la régression linéaire à un sous-domaine de l'espace $\mathbb{R}^d$ (dans le cas de d variables explicatives réelles).

Le chapitre 3 s'intéresse à une recherche des coefficients sur l'espace $\mathbb{R}^d$ tout entier. Cette démarche est naturelle lorsqu'on ignore tout des capacités prédictives des variables explicatives. Toutefois, on verra que la régularisation ℓ_1 étudiée dans ce chapitre vise précisément à restreindre implicitement la zone de recherche (à une "boule ℓ_1").

Le chapitre 4 s'intéresse à des ensembles convexes, bornés mais potentiellement assez grands.

Enfin, les chapitres 5 et 6 présentent le cas de l'agrégation d'experts, où l'on suppose que certaines variables explicatives ont individuellement de bonnes capacités prédictives (d'où le terme "experts"), et où par conséquent des performances a minima correctes sont attendues

sur le simplexe des combinaisons convexes, ce qui n'empêchera pas d'utiliser également des coefficients plus généraux.

La figure 1.2 illustre ces différences dans le domaine de recherche des coefficients.

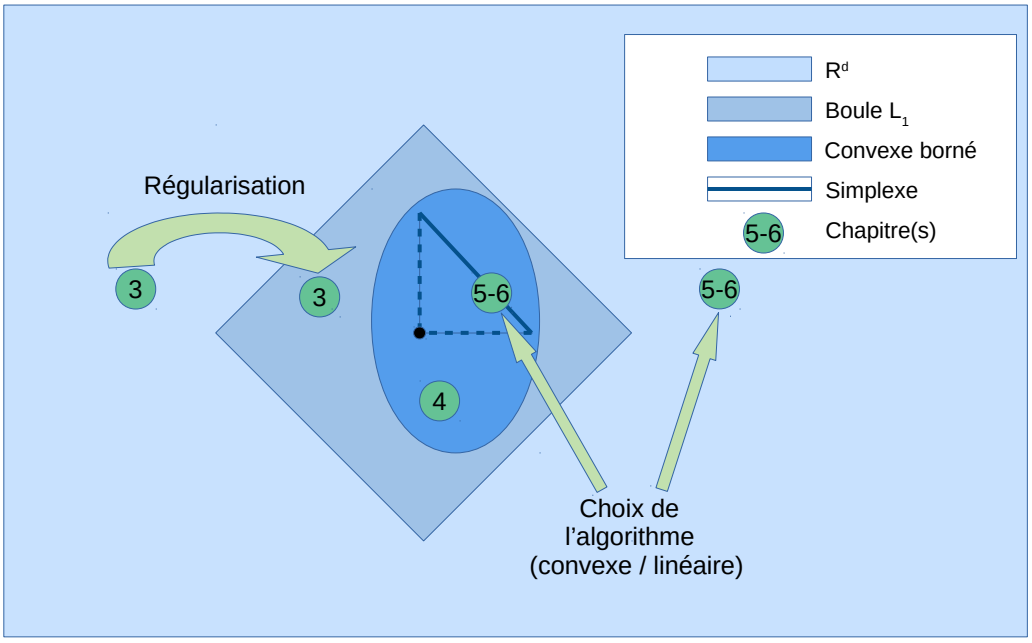


Figure 1.2: Zones de $\mathbb{R}^d$ dans lesquelles sont choisis les coefficients de régression selon les chapitres (la pastille numérotée du chapitre est située dans la/les zone(s) où sont choisis les coefficients)

1.2.2. Un point commun : la mesure des performances est souvent relative

"Faire au mieux avec les données disponibles..."

Les estimateurs de régression linéaire s'appuient par définition sur des prédicteurs (experts, variables explicatives, ...), leurs performances dépendent donc des qualités de prévision de l'ensemble de ces prédicteurs. C'est pourquoi les mesures de performance (et notamment les garanties théoriques) s'effectuent souvent comparativement à un point de référence, un étalon, lié aux prédicteurs. Ce dernier est souvent un oracle, c'est-à-dire qu'il est lié à des quantités inconnues à l'avance.

"... pour ne pas avoir (trop ?) de regret"

La différence entre la perte du statisticien et celle du point de référence est appelée regret (ou, si on parle de son espérance,"excès de risque") :

$$R = \widehat{L}_{\text{algorithme}} - L_{\text{référence}};$$

Il est souvent difficile en théorie de garantir un regret négatif, en particulier dans toutes les situations où le point de référence peut être optimal, et on cherche plutôt dans les résultats théoriques à borner le regret par une quantité positive, la plus faible possible. Le théorème suivant (prouvé dans l'introduction mathématique : voir Corollary 2.3) est un exemple classique pour l'algorithme Exponentially Weighted Average ("EWA"). Cet algorithme convexe repose sur un paramètre d'apprentissage η et ses coefficients $p_{k,t}$ vérifient, dans le cas où l'on dispose de K prédicteurs à agréger : $p_{k,1} = 1/K$ et pour tout instant $t \geqslant 2$:

$$p_{k,t} = \frac{\exp(-\eta L_{k,t-1})}{\sum_{i=1}^{K} \exp(-\eta L_{i,t-1})}.$$

Theorem 1.1. *On note $f_{k,t}$ la prévision du k-ième prédicteur à l'instant t, et on suppose que les pertes des K prédicteurs pour chacun des T instants de prévision sont toutes comprises dans l'intervalle $[m, M]$. Alors l'algorithme "EWA", effectué avec un paramètre d'apprentissage constant $\eta = \sqrt{8\log(K)/\left((M-m)^2 T\right)}$ garantit un regret :*

$$\sum_{t=1}^{T} \ell(\widehat{y}_t^{EWA}, y_t) - \min_{k=1..K} \sum_{t=1}^{T} \ell(f_{k,t}, y_t) \leqslant (M-m)\sqrt{\frac{T\log(K)}{2}}.$$

En revanche, dans les applications pratiques, il arrivera régulièrement qu'on obtienne des regrets négatifs. Ainsi, dans le chapitre 6, lorsqu'on parvient à faire mieux que le meilleur expert, cela signifie qu'on a mieux prédit (en terme d'erreur cumulée) que toutes les simulations fournies par l'IFPEN, autrement dit on a bonifié (en terme de prévisions) les données fournies.

1.3. Chapitre 3 : Obtention d'une régularisation optimale dans un cadre batch stochastique

1.3.1. Cadre mathématique

Présentation générale

Le chapitre 3 de cette thèse s'intéresse à la régularisation, dans un cadre batch stochastique. Dans ce cadre, on cherche à relier les deux variables aléatoires X et Y d'un couple (X, Y) par une fonction $\widehat{g}$ appartenant à un ensemble F, telle que $\widehat{g}(X)$ soit proche de Y (au sens du risque quadratique). On cherche donc à minimiser sur un ensemble F de fonctions le risque quadratique :

$$g \longmapsto \mathbb{E}\left[(Y - g(X))^2\right] \tag{1.3.1}$$

et on s'intéresse au minimiseur g^* correspondant (dont l'existence sera garantie par les hypothèses utilisées). Dans le cas de la régression linéaire, sur lequel nous nous centrons, $g(X) = \langle t, X \rangle$ pour $t \in T \subset \mathbb{R}^d$. Dans le chapitre 3, on ne disposera pas d'information a priori sur T, le vecteur cherché pourra être n'importe quel élément de $\mathbb{R}^d$.

Pour atteindre l'objectif souhaité, on dispose d'un échantillon d'apprentissage de N couples i.i.d. (indépendants et identiquement distribués) : $(X_1, Y_1), \ldots, (X_N, Y_N)$, de même loi que (X, Y). Dans ce chapitre, on recherchera des résultats valables avec grande probabilité par rapport à cet échantillon d'apprentissage.

Design aléatoire

Quand les X_i sont aléatoires, on parle alors de "design aléatoire". Le risque quadratique (1.3.1) d'un algorithme est conditionné par l'aléa de l'échantillon d'apprentissage (à la fois les X_i et les Y_i). Dans le chapitre 3, on s'intéressera à un design gaussien pour les X_i. Ce design est classique et possède la propriété géométrique d'isotropie : si X est un vecteur gaussien standard de $\mathbb{R}^d$, pour tout $t \in \mathbb{R}^d$, $\mathbb{E}\left[\langle t, X \rangle^2\right] = \|t\|_2^2$.

Design fixe

On s'intéressera également dans le chapitre 3 (Section 3.4) au cas du design fixe. Dans ce cas les X_i sont fixés (seuls les Y_i sont aléatoires), et le risque quadratique est alors défini par rapport au design (i.e., conditionnellement aux X_i) et devient :

$$\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \langle t, X_i \rangle\right)^2 \middle| X_1, \ldots, X_N\right].$$

La matrice de design $\mathbb{X}$ dont les lignes successives sont $X_1^\top, \ldots, X_N^\top$, joue un rôle important. Dans la section 3.4 du chapitre 3, on utilisera une hypothèse d'isométrie restreinte (cf. Bickel et al. [2009] and Bühlmann and van de Geer [2011]), garantissant sur un cône de $\mathbb{R}^d$ que tout élément t du cône vérifie :

$$\frac{1}{N}\sum_{i=1}^{N}\left(\langle t, X_i \rangle\right)^2 \simeq \|t\|_2^2.$$

Cette hypothèse fait écho à la propriété d'isométrie d'un design gaussien standard. Ainsi, la méthodologie appliquée pour le design aléatoire du chapitre 1 sera partiellement transposable à la section sur le design fixe, avec néanmoins quelques adaptations nécessaires (voir l'une d'entre elles en fin de section 1.3.5).

1.3.2. Régularisation

Minimiseur du risque empirique

Le minimiseur du risque empirique (ici, l'"estimateur des moindres carrés"), ou "ERM" (Empirical Risk Minimizer), est un candidat naturel :

$$\widehat{t}_{ERM} \in \operatorname*{argmin}_{t \in T} \left\{ \sum_{i=1}^{N} \left(Y_i - \langle t, X_i \rangle \right)^2 \right\}.$$

Cet estimateur a été largement étudié (voir par exemple Stein [1956] pour le fameux paradoxe de Stein).

Sous certaines hypothèses (T convexe, cadre sous-gaussien), Lecué and Mendelson [2013] montrent que l'ERM est optimal (minimax) en déviation sur T. Cela ne veut pas dire qu'il est optimal sur tous les sous-ensembles de T, en particulier ceux de taille beaucoup plus réduites. En effet, lorsque T est très grand, l'ERM a tendance à faire du sur-apprentissage : il "colle" trop à l'échantillon d'apprentissage, se laissant influencer par le bruit et les éventuels données atypiques, et s'éloignant ainsi de la cible théorique

$$t^* \in \operatorname*{argmin}_{t \in T} \mathbb{E} \left[\left(Y - \langle t, X \rangle \right)^2 \right]. \tag{1.3.2}$$

Dans ce qui suit, on fera la supposition (valable presque sûrement, sous certaines hypothèses) que cet argmin est réduit à un singleton t^*.

Si l'on a une idée assez précise de la zone où se situe t^*, une solution consiste bien sûr à restreindre l'ensemble de recherche T à cette zone. Mais dans le cadre du chapitre 3, on ne suppose pas avoir cette connaissance a priori et on prend $T = \mathbb{R}^d$ tout entier.

Régularisation

Une idée très utilisée pour pallier ce problème correspond d'une certaine manière à "régulariser", c'est-à-dire à ajouter au risque empirique d'un vecteur t un "coût" supplémentaire, le terme de régularisation $\Psi(t)$. L'estimateur devient alors un minimiseur du risque empirique régularisé :

$$\widehat{t}_{\Psi} \in \operatorname*{argmin}_{t \in T} \left\{ \sum_{i=1}^{N} \left(Y_i - \langle t, X_i \rangle \right)^2 + \Psi(t) \right\}. \tag{1.3.3}$$

On introduit ainsi un biais favorisant les zones où Ψ est faible (souvent, les zones proches de l'origine).

Une fois défini par (1.3.3) $\widehat{t}_{\Psi}$ vérifie la propriété suivante (qui ne peut être une définition vue sa récursivité) :

Lemma 1.2.

$$\widehat{t}_{\Psi} \in \operatorname*{argmin}_{t \in T:\ \Psi(t) \leqslant \Psi(\widehat{t}_{\Psi})} \left\{ \sum_{i=1}^{N} \left(Y_i - \langle t, X_i \rangle \right)^2 \right\}. \tag{1.3.4}$$

En effet, s'il existait un élément ayant à la fois un risque empirique strictement inférieur à celui de $\widehat{t}_\Psi$ et une régularisation inférieure ou égale à $\Psi(\widehat{t}_\Psi)$, cet élément aurait alors un risque empirique régularisé strictement inférieur à celui de $\widehat{t}_\Psi$, en contradiction avec la définition (1.3.3).

On peut interpréter cette expression (1.3.4) comme le fait qu'un minimiseur du risque empirique régularisé est aussi, implicitement, un minimiseur du risque empirique sur un ensemble (inconnu à l'avance) plus restreint : $\{t : \Psi(t) \leqslant \Psi(\widehat{t}_{RERM})\}$. On verra que tout l'enjeu d'un bon choix de régularisation est précisément que cet ensemble restreint ait une taille adéquate, proche de celle de l'ensemble lié à l'estimateur "oracle" $\{t : \Psi(t) \leqslant \Psi(t^*)\}$.

1.3.3. Apports de la thèse

Les travaux présentés au chapitre 3 définissent d'abord un critère d'optimalité pour les fonctions de régularisation (voir Section 1.3.4).

L'apport principal du chapitre est une "preuve de concept", appliquée à la régularisation ℓ_1 et dans le cadre défini ci-dessus, d'une approche permettant de construire une fonction de régularisation optimale (voir Section 1.3.5 pour les grandes lignes de cette construction).

1.3.4. Définition d'un critère d'optimalité pour la fonction de régularisation

Norme et fonction de régularisation

Le choix du terme de régularisation $\Psi(t)$ est très important. Il est généralement fonction (croissante) d'une norme du vecteur t : $\Psi(t) = g(\|t\|)$. Le choix de cette norme peut venir d'un a priori sur la cible : si l'on sait que l'argmin (1.3.2) est réduit à un singleton t^* tel que $\|t^*\|$ est faible pour une certaine norme $\|\cdot\|$, on utilisera une régularisation fondée sur cette norme . Le choix peut aussi venir d'une volonté du statisticien d'utiliser les propriétés liées à une certaine norme. Ainsi, Tibshirani [1996] introduit la régularisation LASSO, qui utilise le terme de régularisation : $\Psi(t) = \lambda\|t\|_1$ (avec un paramètre λ fixé par le statisticien) et permet souvent, grâce aux propriétés géométriques de la norme $\|.\|_1$, d'obtenir un vecteur parcimonieux ("sparse"). Cela peut se révéler utile, par exemple dans un contexte de haute dimensionnalité ou de données massives ("Big Data"). L'estimateur LASSO est donc :

$$\widehat{t}_{LASSO} \in \underset{t\in\mathbb{R}^d}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \big(Y_i - \langle t, X_i\rangle\big)^2 + \lambda\|t\|_1 \right\}.$$

(on peut aussi, par convention, comme dans le chapitre 3, diviser la quantité à minimiser par N et adapter λ, ce qui bien sûr est sans conséquence). Une autre régularisation classique, la régression Ridge (introduite dans Hoerl [1962], utilisée également par Tikhonov and Arsenin [1977]), utilise, elle, le carré de la norme ℓ_2 : $\Psi(t) = \lambda\|t\|_2^2$. Elle peut parfois surpasser le LASSO en prévision, et peut être utile quand on dispose de variables fortement corrélées et qu'on veut éviter qu'il n'y ait qu'une seule d'entre elles qui soit sélectionnée. Ces éléments poussent Zou and Hastie [2005] à proposer de sommer les régularisations ℓ_1 et ℓ_2, aboutissant à un estimateur nommé "Elastic Net". Sa régularisation est $\lambda_1\|.\|_1 + \lambda_2\|.\|_2^2$ avec $\lambda_1 > 0$ et $\lambda_2 > 0$ éventuellement différents.

Une fois la norme choisie, il reste à décider d'une fonction de régularisation, qui transformera cette norme en un terme de régularisation. On peut dès le départ se poser la question suivante : quelles sont les caractéristiques d'une "bonne" fonction de régularisation ? Le premier apport du chapitre 3 est donc de proposer un critère d'optimalité pour une fonction de régularisation, pour une norme donnée.

Un critère d'optimalité

Comme on l'a vu en (1.3.4), la régularisation aboutit en fait à la combinaison d'une restriction de la région de l'espace $\mathbb{R}^d$ où l'estimateur est cherché, et d'une minimisation du risque empirique sur cette zone restreinte. Lorsque Ψ est fonction croissante d'une norme, les zones restreintes possibles sont les boules (pour la norme considérée) centrées en 0, et de rayons différents. Le modèle optimal, que l'on notera B^*, est la boule centrée en 0 et de rayon $\|t^*\|$ –qui est inconnu. L'ERM restreint à cette boule B^* (qui est donc un modèle oracle) est lui-même minimax sur cette boule B^* (cf. [Lecué and Mendelson, 2013]). On définit alors dans le chapitre 3 un critère d'optimalité pour une fonction de régularisation. Il s'agit pour l'estimateur régularisé $\widehat{t}_\Psi$ de parvenir à obtenir un risque inférieur, à une constante multiplicative absolue près, au risque de cet ERM oracle sur la boule B^* (noté $\widehat{t}^{B^*}_{ERM}$) :

$$\exists\, C :\ \forall\, t^*,\ \mathbb{E}\left[\left(Y - \langle \widehat{t}_\Psi, X\rangle\right)^2\right] \leqslant C\ \mathbb{E}\left[\left(Y - \langle \widehat{t}^{B^*}_{ERM}, X\rangle\right)^2\right] \tag{1.3.5}$$

Tout l'enjeu, pour arriver à remplir ce critère, est donc de définir une fonction de régularisation capable de "sélectionner" une région de l'espace proche du modèle optimal.

Précisons que ce critère est un critère théorique, et qu'une fonction de régularisation le remplissant peut éventuellement être difficile à calculer (non-convexité, etc.)

1.3.5. Construction d'une fonction de régularisation optimale

Cette section décrit les grandes lignes du raisonnement qui fonde le chapitre 3.

Idée directrice

Le chapitre 3 se veut une "preuve de concept" du principe suivant : "pour obtenir une régularisation optimale pour un problème donné, il faut créer une régularisation "sur-mesure" pour le problème en question, en déterminant "les zones où elle est nécessaire, et sa valeur nécessaire sur ces zones". C'est cette idée de régulariser "suffisamment mais pas plus que nécessaire pour le problème étudié", qui guide notre approche.

Dans ce chapitre, nous appliquons ce principe à un cadre classique gaussien, et à la norme de régularisation ℓ_1 (qui est, comme indiqué plus haut, très utilisée pour ses propriétés de parcimonie). Plusieurs données du problème (par exemple la densité –ou même la log-densité– gaussienne) ne sont en rien linéaire vis-à-vis des normes des vecteurs de l'espace, pourquoi donc le choix habituel "LASSO" d'une régularisation linéaire en la norme (de type $\lambda\|.\|_1$) serait-il optimal ? (Effectivement, il ne l'est pas.)

On cherche à obtenir une régularisation permettant de sélectionner, avec grande probabilité, un modèle de taille proche de celle du modèle optimal B^*. Cette régularisation doit donc être :

- suffisante pour exclure les modèles trop grands,
- mais suffisamment faible pour ne pas trop biaiser l'estimateur vers l'origine et continuer à inclure les modèles de taille proche de celle de B^*.

Probabilités et géométrie, localisation

L'essentiel de notre approche consistera à contrôler "au plus juste" (à des constantes absolues multiplicatives près), sur les modèles les plus grands, les processus aléatoires, grâce à la régularisation. Plus précisément, on va garantir (avec grande probabilité) que tout vecteur t vérifiant $\|t\|_1 \geqslant \omega\|t^*\|_1$ (pour une certaine constante absolue $\omega > 1$) a un risque régularisé strictement plus grand que celui de t^*, et ne peut donc être $\widehat{t}$.

Pour cela, on va introduire dans nos raisonnements, pour tout vecteur t, la différence entre le risque empirique régularisé de t et celui de t^*. Cette quantité, qu'on notera $P_N\mathcal{L}_t^\Psi$, est un intermédiaire de calcul inconnu, car dépendant de t^*. Elle vérifie par construction $P_N\mathcal{L}_{\widehat{t}}^\Psi \leqslant P_N\mathcal{L}_{t^*}^\Psi = 0$, et on veut montrer que, sur les grands modèles, tous les vecteurs de trop grande norme (que l'on veut donc exclure) vérifient $P_N\mathcal{L}_t^\Psi > 0$ et ainsi ne peuvent pas être $\widehat{t}$.

On introduit la décomposition suivante :

$$P_N\mathcal{L}_t^\Psi = P_N\mathcal{L}_t + \mathcal{R}_{t,t^*}$$

où $P_N\mathcal{L}_t$ est un processus aléatoire (la différence des risques empiriques de t et t^*) et $\mathcal{R}_{t,t^*} = \psi(t) - \psi(t^*)$ une quantité déterministe (la différence des régularisations en t et t^*), proche de $\Psi(t)$ si $\|t\|_1 \gg \|t^*\|_1$.

Si $\|t\|_1 \geqslant \|t^*\|_1$, $\mathcal{R}_{t,t^*}$ est positif et donc dans ce cas, $P_N\mathcal{L}_t > 0$ suffit à garantir que $P_N\mathcal{L}_t^\Psi > 0$. La régularisation n'est par conséquent "utile" que dans la zone où l'on n'est (sur un événement de grande probabilité défini dans la preuve) pas sûr que $P_N\mathcal{L}_t > 0$. Or, on verra que cette zone est une boule ℓ_2.

On est ainsi amené à considérer des intersections des modèles (des boules ℓ_1) avec cette boule ℓ_2 "où la régularisation peut être nécessaire" : cette approche est nommée "localisation". Elle est représentée en Figure 1.3 (où l'on n'a pas représenté une boule entière $t^* + \rho B_1^d$, avec B_1^d désignant la boule unité fermée de $\mathbb{R}^d$ pour la norme ℓ_1, mais plutôt la couronne $t^* + \rho B_1^d \backslash (\rho/2) B_1^d$ vu que ce sont les vecteurs de grande norme ℓ_1 que l'on souhaite écarter). La fonction de régularisation f présentée dans l'article est précisément choisie de façon à contrôler $P_N\mathcal{L}_t$ (et plus précisément le seul terme de $P_N\mathcal{L}_t$ potentiellement négatif) sur cette intersection.

Le rayon r de la boule ℓ_2 joue un rôle crucial ; on verra qu'on le déterminera à partir de deux équations de points fixes, faisant intervenir la "fenêtre gaussienne" des ensembles localisés :

$$\ell^*(\rho B_1^d \cap r B_2^d) := \mathbb{E}\left[\sup_{t \in \rho B_1^d \cap r B_2^d} \langle G, t \rangle\right]$$

où G est un vecteur gaussien standard de $\mathbb{R}^d$.

Le théorème principal du chapitre 3 (théorème 3.1.4) introduit une fonction de régularisation judicieusement choisie (dépendant notamment des deux équations de points fixes évoqués ci-dessus), et montre qu'elle est optimale pour le critère (1.3.5). Les quantités en jeu dans (1.3.5) dépendent elles aussi des deux équations de points fixes.

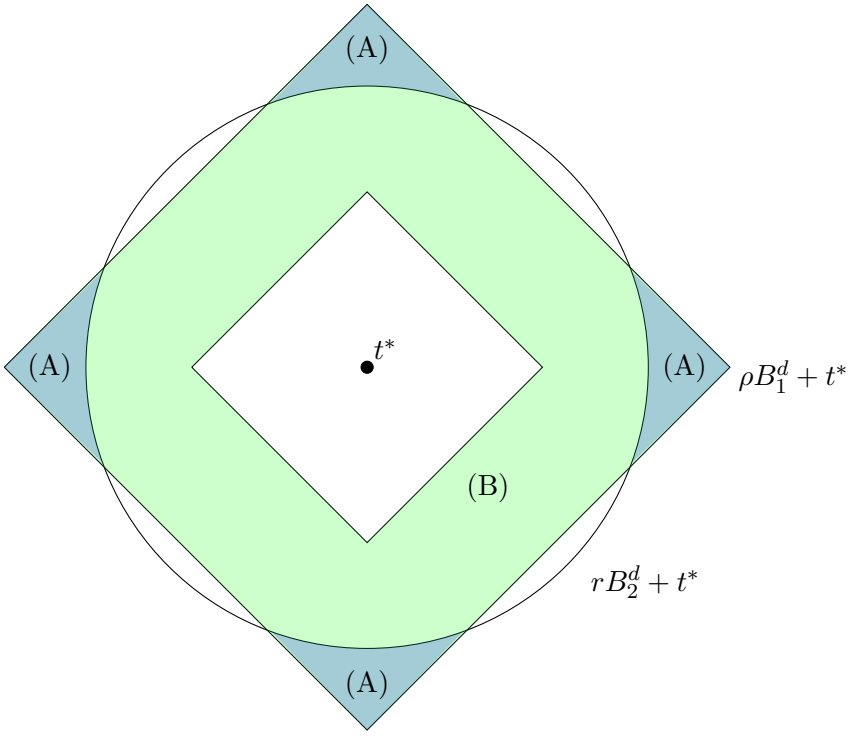


Figure 1.3: (A) : Régularisation non nécessaire. (B) : La régularisation peut être nécessaire.

Une idée importante pour le design fixe

Dans le cas du design fixe, la matrice des variables explicatives $\mathbb{X}$, de taille $N \times d$ et dont les lignes successives sont $X_1^\top, \ldots, X_N^\top$, est déterministe.

L'idée-clé permettant l'adaptation de la preuve dans le cas de ce design fixe est le transfert de la localisation : on passe d'une localisation dans l'espace de départ $\mathbb{R}^d$ (dans la partie "design aléatoire") à une localisation dans l'espace d'arrivée $\mathbb{R}^N$ (dans la partie "design fixe"). Ce choix est finalement assez naturel, car en termes de prévisions ce ne sont pas les vecteurs t qui interviennent, mais uniquement leur image $\mathbb{X}t$ par la matrice déterministe $\mathbb{X}$. On intersecte ainsi, non pas le modèle ρB_1^d mais son image $\rho \mathbb{X} B_1^d \subset \mathbb{R}^N$, par une boule ℓ_2 de $\mathbb{R}^N$, en l'occurrence rB_2^N.

1.3.6. Pistes de recherche futures

Ce chapitre se veut une "preuve de concept" d'une approche pour construire des régularisations optimales, aussi une première piste serait-elle de l'appliquer sous d'autres hypothèses. En particulier, on pourrait changer la géométrie du problème :

- en remplaçant les hypothèses permettant une certaine isométrie ℓ_2 (design gaussien, hypothèse d'isométrie restreinte) ;
- en utilisant une autre norme que la norme ℓ_1.

S'agissant de l'estimateur présenté dans ce chapitre, une étude de ses propriétés de parcimonie (par exemple comparativement au LASSO) serait judicieuse.

Enfin, sa mise en pratique (non triviale, car nécessitant une optimisation non convexe) sur des jeux de données réels pourrait être riche d'enseignements.

1.4. Chapitre 4 : Améliorations d'un algorithme adaptatif d'optimisation convexe séquentielle : MetaGrad

1.4.1. Optimisation convexe séquentielle

Présentation de l'optimisation convexe séquentielle

Dans le chapitre 4, on s'intéresse à un problème un peu plus général que la prévision : l'optimisation convexe séquentielle ("Online Convexe Optimization"), dans $\mathbb{R}^d$. Dans les autres chapitres, on cherchait à minimiser l'erreur de prévision :

$$\widehat{w} \mapsto \ell(y_{\text{observé}} - \widehat{w}^\top x)$$

avec ℓ une fonction de perte, par exemple la fonction carré (perte quadratique), $\widehat{w}$ le vecteur de poids et x le vecteur des variables explicatives. On peut voir la quantité précédente comme un cas particulier d'une perte générale, potentiellement dépendante du temps, $\widehat{w}_t \mapsto \ell_t(\widehat{w}_t)$. Lorsque ℓ_t est convexe pour tout t, et qu'on cherche à minimiser la perte cumulée :

$$\sum_{t=1}^{T} \ell_t(\widehat{w}_t)$$

on parle d'optimisation convexe séquentielle. On est alors à mi-chemin entre la prévision séquentielle statistique, et l'optimisation convexe, et on peut tirer profit d'idées des deux domaines. En particulier, cette vision "optimisation" apporte une vision "spatiale" du problème : on cherche à "se déplacer vers un optimum". Deux cadres seront étudiés : celui déterministe (type "suites individuelles") où le minimum est susceptible de changer à chaque tour ; et un cadre stochastique dans lequel le minimum est fixe mais où les pertes sont bruitées.

Domaines d'étude

Dans le cas du chapitre 4, on s'intéressera à des domaines convexes, bornés (ou à certains moments "faiblement bornés", au sens où le produit scalaire de leurs vecteurs avec les données sera borné) —mais potentiellement grands. Si cette limitation du domaine est une hypothèse utile pour certaines démonstrations, et si elle permet (lorsque le domaine est judicieusement choisi) d'améliorer les performances pratiques en évitant aux prévisions de "s'égarer", elle peut avoir néanmoins un inconvénient : elle entraîne bien souvent dans les algorithmes une étape supplémentaire de projection sur le domaine. En effet, certains algorithmes aboutissent parfois à un élément $\tilde{x}$ hors du domaine K, et demandent ensuite d'appliquer une projection de type :

$$\widehat{x} = \operatorname*{argmin}_{x \in K} d(x, \tilde{x})$$

le choix de la distance d dépendant de l'algorithme. Cette projection peut s'avérer coûteuse en terme de temps de calcul.

L'utilisation de domaines "faiblement bornés" dans le chapitre est ainsi motivée par l'existence d'une formule explicite permettant une projection rapide. On peut aussi appliquer des techniques inspirées de l'optimisation : descente de gradient, méthode de Newton modifiée, etc., en les adaptant au problème.

Importance de la perte

Les méthodes et performances optimales dépendent des hypothèses sur les pertes ℓ_t, notamment leur courbure. La convexité garantit une certaine sécurité dans le comportement des algorithmes, mais sous des hypothèses de courbure plus fortes, on peut obtenir des performances meilleures. L'hypothèse de stricte convexité est peu utilisée sur le plan théorique (elle permet de garantir l'unicité du minimum, mais apporte peu sur la convergence vers ce minimum, vu que son "écart" avec la convexité simple n'est pas quantifié). En revanche, deux hypothèses plus fortes permettent d'obtenir de meilleurs résultats. Tout d'abord, l'(α-)exp-concavité : la perte ℓ_t est dite α-exp-concave (pour $\alpha > 0$) si $x \mapsto \exp\left(-\alpha \ell_t(x)\right)$ est concave. Cette hypothèse garantit en particulier que ℓ_t est convexe. Une hypothèse encore plus forte est la (λ-)convexité forte : pour $\lambda > 0$, la perte ℓ_t est dite λ-fortement convexe si :

$$\forall x, y, \quad \ell_t(y) \geqslant \ell_t(x) + \left\langle \nabla \ell_t(x), y - x \right\rangle + \frac{\lambda}{2} \|y - x\|_2^2.$$

Enjeux autour de la perte et de la calibration des paramètres

Des méthodes différentes ont été étudiées pour obtenir des performances optimales selon chacune des hypothèses ci-dessus, et choisir judicieusement l'une ou l'autre d'entre elles demande donc de déterminer la courbure de la perte. Un enjeu de recherche important est donc de déterminer des méthodes adaptatives, c'est-à-dire des méthodes avec des garanties optimales ou quasi-optimales sous différents jeux d'hypothèses (par exemple à la fois optimales sous l'hypothèse de convexité simple, et sous l'hypothèse d'exp-concavité). En plus des hypothèses sur la perte, une autre question d'importance qui se pose au statisticien est la calibration des paramètres de ses algorithmes lorsque ces derniers en comportent. Cette question rejoint

d'ailleurs la précédente lorsque le paramètre optimal pour l'algorithme dépend du paramètre α d'exp-concavité ou λ de convexité forte, potentiellement inconnus.

1.4.2. Algorithme MetaGrad, apports de la thèse

Présentation générale de l'algorithme MetaGrad

L'algorithme MetaGrad (van Erven and Koolen [2016]) étudié et amélioré dans cette thèse est précisément motivé par les deux objectifs précédents et propose des avancées sur chacun d'eux. Il s'agit en fait d'un méta-algorithme, combinant (via une agrégation proche des poids exponentiels) des algorithmes auxiliaires qui sont des versions modifiées de la méthode "Online Newton Step" (introduite dans Hazan et al. [2007]). Chaque algorithme auxiliaire utilise un paramètre d'apprentissage différent (d'où le nom MetaGrad, pour "Multiple Eta Gradient"), et le méta-algorithme maître parvient à obtenir des garanties proches de celle du meilleur algorithme auxiliaire.

L'algorithme MetaGrad parvient à supprimer le problème de la calibration du paramètre d'apprentissage (il contient d'autres paramètres : hyper-paramètres de grille et paramètres d'initialisation, mais ceux-ci sont beaucoup plus faciles à calibrer). Il parvient également dans le cadre de l'optimisation convexe séquentielle "déterministe", à obtenir, en ce qui concerne la dépendance en la dimension d et le nombre T des prévisions, des bornes quasi-optimales sous hypothèses de convexité simple, et optimales (à constante multiplicative près) sous hypothèses d'exp-concavité.

Accélérations de MetaGrad

MetaGrad possède néanmoins un point faible : il est (relativement) lent, a fortiori en grandes dimensions. L'un des axes de recherche consiste à proposer des versions modifiées, plus rapides mais aux performances et garanties proches. Un premier exemple était présenté dans l'article originel (une version diagonale de l'algorithme), dans cette thèse nous proposons plusieurs autres versions modifiées, inspirées notamment du "sketching" de Luo et al. [2016], que l'on présente ci-dessous, et nous étudions leurs garanties théoriques.

L'algorithme MetaGrad nécessite à chaque instant t le calcul d'une matrice de covariance de taille $d \times d$: $(\varepsilon I_d + G_t^\top G_t)^{-1}$, où G_t est une matrice de taille $t \times d$. La formule de Woodbury permet d'écrire :

$$(\varepsilon I_d + G_t^\top G_t)^{-1} = \frac{1}{\varepsilon}\left(I_d - G_t^\top\left((\varepsilon I_t + G_t G_t^\top)^{-1} G_t\right)\right).$$

Le sketching consiste à remplacer $G_t^\top G_t$ par une approximation de rang moindre, pour accélérer les calculs. Plus précisément, on va chercher à introduire une matrice S_t de taille $m \times d$ vérifiant les conditions suivantes :

- $m < t$ et, autant que possible, $m \ll d$,
- $S_t^\top S_t$ est une "bonne" approximation de $G_t^\top G_t$ (pour des critères qui seront présentés dans le chapitre 4)

- S_t et $(\varepsilon I_t + S_t S_t^\top)^{-1}$ sont calculables facilement (par mise à jour par exemple).

L'approche consiste alors à remplacer dans les calculs G_t par S_t et $(\varepsilon I_d + G_t^\top G_t)^{-1}$ par :

$$(\varepsilon I_d + S_t^\top S_t)^{-1} = \frac{1}{\varepsilon}\left(I_d - S_t^\top \left((\varepsilon I_t + S_t S_t^\top)^{-1} S_t\right)\right)$$

ce qui accélère fortement les calculs correspondants.

Une piste pour le futur consisterait à chercher d'autres "accélérations" possibles pour MetaGrad, et à les tester sur des jeux de données en s'intéressant à la fois à leurs améliorations en termes de temps de calculs, et à l'éventuelle dégradation (ou, qui sait, amélioration) de leurs performances pratiques par rapport au MetaGrad initial.

Conversion online-to-batch "améliorée" de MetaGrad

Un autre axe de travail présenté consiste à adapter l'algorithme MetaGrad, conçu pour un cadre séquentiel déterministe, au cadre batch stochastique. La méthode de conversion "online-to-batch" classique, appliquée à MetaGrad, permet d'obtenir sous des hypothèses d'exp-concavité et de convexité forte une erreur dont l'espérance est de l'ordre de $d\log(T)/T$, alors que les ordres de grandeur optimaux sont respectivement pour ces deux hypothèses de d/T et $1/T$. En nous inspirant de Hazan and Kale [2014], nous proposons une conversion "online-to-batch" plus performante, garantissant une erreur de l'ordre de $d\log\log(T)/T$ –mais pour l'instant sous l'hypothèse de convexité forte.

Cette conversion utilise une division de l'échantillon d'apprentissage en périodes ("epochs"), dont la taille suit une croissance géométrique : ainsi, la taille T_j de la période j vérifie $T_j = 2^j T_0$. Sur chaque période, l'algorithme MetaGrad est appliqué, avec à chaque fois une mise à jour des paramètres d'initialisation (plus pertinente dans le cas de MetaGrad que la mise à jour du paramètre d'apprentissage utilisée par Hazan and Kale [2014]). Le point de départ x_1^{j+1} de la période $j+1$ est la moyenne des vecteurs produits par l'algorithme sur la période précédente :

$$x_1^{j+1} = \frac{1}{T_j}\sum_{t=1}^{T_j} x_t^j$$

où x_t^j désigne le vecteur produit par l'algorithme au t-ième instant de la période j.

L'analyse de l'algorithme MetaGrad permet d'obtenir une borne sur le regret dépendant uniquement de paramètres déterministes (paramètres du problème, paramètres de l'algorithme sur cette période) et de la distance $\|x_1^j - x^*\|_2$ entre le point de départ et la cible :

$$\mathbb{E}\big[F(x_1^{j+1})\big] - F\big[x^*\big] \leqslant g_1\left(\|x_1^j - x^*\|_2,\ \text{paramètres déterministes}\right)$$

pour une certaine fonction g_1, avec F la fonction qu'on cherche à minimiser (espérance des pertes), et x^* son minimum.

Une hypothèse de convexité forte permet ensuite de relier $\|x_1^j - x^*\|_2$ au regret correspondant $\mathbb{E}\big[F(x_1^j)\big] - F\big[x^*\big]$, et ainsi obtenir une récurrence :

$$\mathbb{E}\big[F(x_1^{j+1})\big] - F\big[x^*\big] \leqslant g_2\left(\mathbb{E}\big[F(x_1^j)\big] - F\big[x^*\big]\right)$$

pour une fonction g_2 liée à g_1.

Cette récurrence permet de garantir, pour une constante C liée aux données du problème :

$$\mathbb{E}\big[F(x_1^j)\big] - F\big[x^*\big] \leqslant C\frac{\log\big(\log(T_j)\big)}{2^j}.$$

Elle aboutit (moyennant notamment le calcul de la taille minimale nécessaire pour la première période, ce qui introduit en particulier un facteur d de dimension) au théorème suivant, tiré du théorème 4.13.

Theorem 1.3. *Soient $f_1, \ldots, f_T$ des fonctions aléatoires tirées de manière i.i.d., fortement convexes, et définies sur un sous-ensemble convexe $\mathcal{K}$ de $\mathbb{R}^d$. On suppose que l'on connaît une borne uniforme G sur les (sous-)gradients des f_t, et une borne D sur le diamètre de $\mathcal{K}$; mais le paramètre λ de convexité forte des f_t n'est pas supposé connu.*

Soit $F : x \in \mathcal{K} \longmapsto \mathbb{E}[f_1(x)]$, dont on suppose qu'elle admet un minimum :

$$x^* = \operatorname*{argmin}_{x\in\mathcal{K}} F(x).$$

Alors le vecteur $\widehat{x}_T$ résultant de l'application de l'algorithme "Epoch MetaGrad" vérifie, en ce qui concerne la dépendance en d et T :

$$\mathbb{E}\left[F(\widehat{x}_T) - F(x^*)\right] = O\left(\frac{d\log\log(T)}{T}\right).$$

Si on connaît le paramètre de convexité forte λ, on peut utiliser l'algorithme "Epoch Online Newton Step" présenté au chapitre 4 qui garantira une vitesse en $O(d/T)$: voir Théorème 4.9. On pourra également, comme signalé en fin de chapitre 4, utiliser une version modifiée d'"Epoch MetaGrad" qui utilise explicitement λ mais garantit une vitesse en $O(d/T)$.

Une piste de recherche pour le futur serait de chercher à adapter l'analyse dans le but de montrer que cette borne en $d\log\log(T)/T$, voire même en d/T, reste valable sous l'hypothèse plus faible d'exp-concavité. On pourra aussi chercher à supprimer le terme en d dans le cas de la convexité forte, mais cet objectif paraît plus difficile.

1.4.3. Pistes de recherches futures

Il serait judicieux de tester l'algorithme MetaGrad et ses différentes déclinaisons, dont celles présentées dans cette thèse, sur différents jeux de données réelles, à la fois pour juger des performances, mais aussi pour comparer les temps de calcul.

Comme indiqué, la section sur la conversion "online-to-batch" de MetaGrad se veut une première étape vers la construction, dans un cadre batch stochastique, d'un algorithme optimal sous l'hypothèse d'exp-concavité. Il serait ainsi intéressant de tenter de mener à son terme la résolution de ce problème.

1.5. Chapitre 5 : Obtention de faisceaux de prévision par une approche d'agrégation séquentielle

1.5.1. Problématique

Une problématique classique...

Le chapitre 5 trouve son origine dans une demande opérationnelle : lorsqu'on dispose d'observations jusqu'à l'instant présent T_0, mais aussi de prévisions d'experts jusqu'à un futur assez lointain T, on souhaite pouvoir fournir des prévisions à long terme. L'incertitude plus grande due à l'éloignement des échéances, amène à proposer plutôt des régions de confiance, des faisceaux de prévision (dans notre cas, ces faisceaux seront des ensembles d'intervalles). Même si l'on va effectuer plusieurs prévisions, il s'agit d'une situation "batch" car on les effectue en une seule fois, sans retour d'expérience intermédiaire.

Formellement, on a donc accès :

- aux observations $y_t, t \leqslant T_0$;
- aux prévisions des experts $f_{j,t}$ à la fois pour $t \leqslant T_0$ et pour $T_0 < t \leqslant T$

et on cherche à prévoir des intervalles de prévision $[\widehat{y}_t^{\min}, \widehat{y}_t^{\max}]$ pour $T_0 < t \leqslant T$.

Cet objectif est relativement classique dans un cadre habituel de séries temporelles, où l'on peut obtenir des intervalles de confiance valables avec une probabilité donnée, par différentes méthodes (calcul direct, simulations de type Monte-Carlo,...) : citons Weron [2002] et Wan et al. [2014] (avec une utilisation de réseaux de neurones).

...abordée sous un angle original

Toutefois on cherche ici à conserver l'esprit "suites individuelles" en ne s'appuyant pas sur une modélisation stochastique, mais uniquement sur les prévisions des experts. En particulier, la notion de "niveau de confiance" (par exemple 95%) n'a pas forcément de sens.

On cumule ainsi les difficultés des suites individuelles (l'absence de modélisation), et celles de la prévision batch (pas de retour d'expérience). Ce problème est donc difficile ; et à notre connaissance il n'a pas encore été abordé dans le cadre des suites individuelles, dont les algorithmes ont crucialement besoin (généralement dans leur définition même) des retours d'expérience, des observations jusqu'à l'instant $t-1$ pour prévoir l'instant t. Or ici, en disposant des observations uniquement jusqu'à un instant T_0 (et, au-delà, seulement des prévision d'experts), on cherche à prévoir non seulement pour T_0+1, mais aussi T_0+2, T_0+3,...

1.5.2. Apports des travaux de cette thèse

Sur ce problème nouveau, l'apport des travaux de cette thèse est triple. Tout d'abord, est proposée une méthodologie pour adapter les algorithmes de suites individuelles à ce cadre. Cette méthodologie nécessitant une optimisation délicate, dépendant fortement de l'algorithme utilisé, on propose deux méthodes pour effectuer cette optimisation : une pour l'algorithme

EWA (avec une extension pour l'algorithme "Fixed-Share EWA"), l'autre pour l'algorithme Ridge. Enfin, ces deux algorithmes sont mis en œuvre (au chapitre 6) sur un jeu de données pétrolier d'IFP Energies nouvelles (avec l'introduction d'un indicateur de performance et d'un point de référence adéquats pour les évaluer).

Méthodologie : "Retour vers le futur"

La méthodologie proposée repose sur l'idée suivante : "Les algorithmes de suites individuelles sont conçus pour être robustes et capables de gérer correctement tous les cas possibles... alors soumettons-leur tous les cas possibles et voyons leurs différentes prévisions".

La méthodologie consiste donc tout d'abord à définir un ensemble $\mathcal{E}$ de "scénarios possibles" :

$$\mathcal{E} = \{y_{T_0+1}, y_{T_0+2}, \ldots, y_T : [\text{certaines conditions}]\}.$$

Cette ensemble doit être suffisamment large. Dans notre étude pratique, nous avons simplement limité les variations entre deux instants consécutifs, créant ainsi un "cône des scénarios possibles" :

$$\mathcal{E} = \{y_{T_0+1}, y_{T_0+2}, \ldots, y_T : \forall\, j \leqslant T - T_0,\ y_{T_0} + j\,\Delta_1 \leqslant y_{T_0+j} \leqslant y_{T_0} + j\,\Delta_2\} \tag{1.5.1}$$

avec Δ_1 et Δ_2 respectivement les bornes inférieure et supérieure des incréments.

Il faut ensuite appliquer l'algorithme sur chaque scénario possible $y_{T_0+1}, y_{T_0+2}, \ldots, y_T \in \mathcal{E}$, et noter les prévisions correspondantes : $\widehat{y}_{T_0+1}, \widehat{y}_{T_0+2}, \ldots, \widehat{y}_T$. Le faisceau de prévisions sera alors défini comme l'ensemble convexe des prévisions obtenues pour les différents scénarios.

On a donc, en quelque sorte, remplacé la "variabilité probabiliste" des approches traditionnelles par une "variabilité ensembliste".

Une optimisation délicate

La méthodologie présentée consiste non pas à calculer les réponses de l'algorithme à chacun des scénarios possibles (il y en a une infinité), mais à calculer pour chaque instant $t = T_0 + k$ le minimum $\widehat{y}_t^{\min}$ et le maximum $\widehat{y}_t^{\max}$ que peut produire l'algorithme à l'instant t sur l'ensemble des scénarios possibles (entre $T_0 + 1$ et $t - 1$).

Il s'agit donc d'une optimisation en dimension $t - T_0 - 1$ (ce qui devient vite grand) d'une fonction souvent hautement non linéaire. Elle est non triviale : on se convaincra facilement de sa difficulté, par exemple sur l'algorithme EWA, en voyant qu'un scénario qui maximise la prévision à l'instant t est un scénario qui attribue un grand poids aux experts qui prévoient une haute valeur à l'instant t ; mais ces experts peuvent très bien avoir prévu des valeurs très différentes sur les instants précédents, ce qui empêche qu'ils aient tous été performants et obtiennent de grands poids. De même, un scénario qui prévoit des valeurs hautes d'observation jusqu'à l'instant $t - 1$ ne va pas forcément aboutir à une prévision haute à l'instant t, cela dépend aussi des prévisions des experts.

La difficulté de cette optimisation dépend bien entendu de l'algorithme utilisé. Nous proposons dans le chapitre 5 un protocole de mise en œuvre dans le cas de deux algorithmes, pour lesquels on dispose d'une forme explicite des poids :

- pour l'algorithme Ridge, un calcul direct permet de résoudre exactement le problème d'optimisation ;
- pour l'algorithme EWA (et sa variante "Fixed-Share"), une méthode itérative aboutit à un intervalle proche de l'intervalle cherché, et qui le contient.

Voici l'expression des faisceaux obtenus pour les faisceaux tirés de l'algorithme Ridge (Lemme 5.2).

Lemma 1.4. *On suppose que l'on dispose de bornes sur les observations :* $\underline{B}_s \leqslant y_s \leqslant \overline{B}_s$ *pour tout instant* $T_0 < s \leqslant t$. *On note* $F_t = (f_{s,j})_{s\leqslant t; j\leqslant K}$ *la matrice des prévisions des experts jusqu'à l'instant* t, *et* $V^{t+1} := (f_{1,t+1}, \dots, f_{K,t+1})(F_t^\top F_t + \lambda I_K)^{-1} F_t^\top \in \mathbb{R}^t$.

Alors l'intervalle de prévisions $[\widehat{y}_t^{\min}, \widehat{y}_t^{\max}]$ *vérifie :*

$$\widehat{y}_t^{\max} = \sum_{s=0}^{T_0} V_s^{t+1} y_s + \sum_{s=T_0+1}^{t} V_s^{t+1} (\overline{B}_s \mathbf{1}_{V_s^{t+1}\geqslant 0} + \underline{B}_s \mathbf{1}_{V_s^{t+1}<0})$$

et

$$\widehat{y}_t^{\min} = \sum_{s=0}^{T_0} V_s^{t+1} y_s + \sum_{s=T_0+1}^{t} V_s^{t+1} (\underline{B}_s \mathbf{1}_{V_s^{t+1}\geqslant 0} + \overline{B}_s \mathbf{1}_{V_s^{t+1}<0}).$$

Comme indiqué en (1.5.1), dans le cadre de l'étude pratique, les bornes considérées seront $\underline{B}_t = y_{T_0} + (t - T_0)\Delta_1$ et $\overline{B}_t = y_{T_0} + (t - T_0)\Delta_2$, formant un "cône des scénarios possibles".

Mise en œuvre pratique

Dans le chapitre 6, nous mettons en œuvre la méthodologie et l'optimisation décrites ci-dessus sur un jeu de données d'IFP Energies nouvelles (cette étude est présentée plus en détails en Section 1.6). Plusieurs adaptations et améliorations pratiques sont utilisées afin d'améliorer la qualité des faisceaux. Par exemple, le niveau de bruit (on s'autorise ici une petite hypothèse de modélisation : on suppose que ce niveau n'augmente pas sur la période de prévision) doit être pris en compte et constituer une largeur minimale pour le faisceau de prévisions. Pour Ridge, une pré-sélection (automatisée) des experts permet également d'améliorer la prévision.

Le passage d'une évaluation qualitative "à l'œil nu" à une évaluation quantitative nécessite de définir un indicateur de la qualité d'un faisceau. La simple proportion d'observations contenues dans le faisceau (qui correspondrait d'une certaine manière à un niveau de confiance : 90%, 85%...) n'est pas pertinente : il suffirait de prendre un faisceau aussi large que possible pour remplir ce critère. La qualité d'un faisceau dépend donc aussi de sa "largeur". On définit donc au chapitre 6 un nouvel indicateur, "l'efficacité" (*"efficiency"*), comme le rapport du nombre d'observations contenues dans le faisceau, à l'aire du faisceau. Il s'agit d'un critère relatif, en ce qu'il n'a de sens que pour comparer des méthodes sur un jeu de données précis.

Une fois ce critère établi, juger de la qualité de nos méthodes nécessite un point de comparaison, un "benchmark" (banc d'essai).

Nous en proposons un (le meilleur faisceau d'experts parmi les choix d'experts "raisonnables", cf. chapitre 6). Nous sommes alors en mesure d'évaluer nos algorithmes. Les résultats obtenus

sont souvent satisfaisants pour Ridge : en dehors des propriétés chaotiques, on parvient généralement à faire aussi bien voir beaucoup mieux que le benchmark, et obtenir des faisceaux souvent précis, avec cependant une marge de progression sur la calibration des paramètres.

1.5.3. Pistes de recherche futures

Sur le plan pratique, améliorer et rendre plus robuste la calibration des paramètres des algorithmes (qui est l'une des principales difficultés rencontrées au chapitre 6) serait une avancée importante.

Un autre travail pourrait consister à chercher comment effectuer l'optimisation évoquée plus haut pour d'autres algorithmes, par exemple LASSO. En ce qui concerne EWA, une optimisation directe et exacte serait également une avancée. L'application de ces algorithmes de faisceaux sur d'autres jeux de données serait également un travail intéressant, permettant notamment de se faire une idée plus complète de leur précision mais aussi de leur robustesse.

Enfin, une piste conséquente de recherche serait d'essayer d'obtenir des garanties théoriques, absolues ou relatives, sur les faisceaux. Cela passera sans doute par la définition d'un critère adapté. En effet, le simple critère de proportion d'observations situées à l'intérieur du faisceau, ou le critère d'efficacité décrit plus haut, qui en découle, ne semblent pas pertinents (il suffit de laisser une "bande" d'observations possibles au-dessus ou au-dessus du faisceau pour s'exposer à voir toutes les observations dans cette "bande" et aucune dans le faisceau). Des pistes envisageables seraient de s'intéresser à des critères de taille du faisceau, de distance des observations par rapport au milieu ou aux limites du faisceau.

1.6. Chapitre 6 : Application de méthodes d'agrégation à la prévision pétrolière

1.6.1. Différentes approches pour la prévision de production pétrolière

Le chapitre 6 est consacré à l'application de méthodes d'agrégation en suites individuelles à la prévision de production pétrolière.

On s'intéresse à la prévision de trois quantités liées aux puits d'un champ pétrolifère (le jeu de données est présenté plus en détails Section 1.6.2) : le débit de pétrole, le débit d'eau et la pression en fond de puits.

Calage d'historique

La prévision de ces quantités s'appuie sur des modèles géophysiques qui tiennent compte notamment des caractérisitiques du sous-sol (perméabilité et porosité des différentes roches, etc), souvent en le subdivisant par une grille. Plusieurs types de données existent pour obtenir des informations sur le sous-sol : les données statiques (données diagraphiques, données sismiques...), et les données dynamiques, qui portent notamment sur la composition et les flux des fluides présents. Les grandeurs que nous chercherons à prévoir font partie de ces données dynamiques.

Un premier type de travail peut consister à utiliser les données statiques pour reconstituer certaines caractéristiques du sous-sol, ce qui permet ensuite d'effectuer un travail de prévision des données dynamiques. Mais il est évidemment souhaitable, pour améliorer la pertinence et la précision des modèles, de tenir compte des données dynamiques dont on a connaissance au fur et à mesure du temps pour améliorer les modèles. Ce problème inverse est nommé "calage d'historique" : il consiste à reconstituer certaines caractéristiques du sous-sol et des fluides présents, de manière à pouvoir expliquer au mieux les données dynamiques observées. C'est un problème très complexe, et coûteux en temps de calcul. De plus, il est "mal posé", au sens où plusieurs sous-sols différents (i.e., plusieurs jeux de paramètres différents pour les divers points de la grille de subdivision du sous-sol) peuvent aboutir aux mêmes grandeurs observées.

Un certain nombre de travaux récents se concentrent davantage sur la capacité prédictive des modèles que sur la "véracité" des paramètres sous-jacents.

Approche des travaux de la thèse

L'utilisation des algorithmes d'agrégation présentée au chapitre 6 propose un compromis intéressant : calculer une fois pour toutes (à partir des données statiques) différents modèles de sous-sol crédibles et les prévisions correspondantes, et agréger ces différentes prévisions en tenant compte à chaque pas de temps des données dynamiques reçues entre-temps. Les algorithmes d'agrégation proposés ont un temps de calcul négligeable devant les calculs géophysiques de prévision, et permettent ainsi de tenir compte au fil du temps des informations fournies par les données dynamiques, sans avoir à recalculer les modèles géophysiques ou leurs prévisions. Il faut toutefois garder à l'esprit que, du fait de la non-linéarité des phénomènes étudiés, l'estimateur agrégé obtenu n'a pas d'interprétation immédiate en termes de composition du sous-sol (moyenner les paramètres de différents modèles de sous-sol considérés, n'est pas équivalent à moyenner les prévisions correspondantes).

La figure 1.4 résume les différentes approches évoquées.

1.6.2. Jeu de données étudié

La figure 1.5 résume les étapes de la construction des prévisions.

Le jeu de données étudié, nommé "Brugge", est artificiel, il a été créé (cf. Peters et al. [2010]) pour servir d'étalon en préparation d'un "workshop" (à Bruges –Brugge en flamand et anglais, d'où son nom) et sert depuis de référence sur ce sujet. Il simule un champ pétrolifère ressemblant à ceux présents en mer du Nord, composé de 20 puits producteurs et de 10 puits injecteurs sur une surface d'une trentaine de km^2, et porte sur des mesures mensuelles sur une période de 10 ans.

Les créateurs de ce jeu de données ont d'abord simulé un modèle très fin (non transmis) de sous-sol, à partir desquelles ils ont pu calculer les données statiques et dynamiques. Ils ont ensuite généré 104 modèles géologiques de sous-sol en adéquation avec les données statiques de ce champ pétrolier. Les caractéristiques de ces modèles, transmises par les créateurs de "Brugge", sont les faciès et divers paramètres (porosité, perméabilité, saturation en eau...) en chaque bloc d'une subdivision du sous-sol par une grille d'environ 60 000 blocs ($139 \times 48 \times 9$).

Approche prédictive classique	**Approche proposée dans cette thèse**	**Approche habituelle pour le calage d'historique**
Les modèles sont calculés en fonction des données statiques.	Les modèles sont calculés en fonction des données statiques.	Les modèles sont mis à jour en fonction des données dynamiques (en plus des données statiques).
Les prévisions sont effectuées une fois pour toutes.	Les prévisions sont agrégées en tenant compte des données dynamiques jusqu'à l'instant précédent.	Les prévisions tiennent donc compte des données dynamiques qui sont obtenues.
Approche peu coûteuse en temps de calcul, mais ne tient pas compte des retours d'expérience au fil du temps.	Approche peu coûteuse en temps de calcul, et utilisation des retours d'expérience au fil du temps.	Approche complète, mais complexe et très coûteuse en temps de calculs.

Figure 1.4: La méthodologie proposée dans cette thèse propose un compromis intéressant entre plusieurs approches existantes.

Ces caractéristiques ont été transmises par les créateurs de "Brugge", ainsi qu'une version bruitée des propriétés dynamiques à prévoir afin de pouvoir juger de l'efficacité des méthodes utilisées. Ces propriétés sont :

- la pression en fond de puits pour les puits injecteurs et producteurs ;
- le débit d'eau pour les puits producteurs ;
- le débit de pétrole pour les puits producteurs.

A partir de chacun de ces 104 modèles, les chercheurs d'IFP Energies nouvelles (IFPEN) Sébastien Da Veiga et Véronique Gervais-Couplet, ont calculé des prévisions mensuelles sur 10 ans pour ces propriétés dynamiques. Ce sont ces prévisions qui sont utilisées au chapitre 6, ainsi que les versions bruitées des données dynamiques.

1.6.3. Apports de la thèse

Prévision ponctuelle

Au chapitre 6, on applique trois algorithmes : EWA (voir Section 1.2.2), Ridge et LASSO (voir Section 1.3.4) , pour agréger sur chacune des propriétés étudiées, les prévisions liées aux 104 modèles de sous-sol, mois par mois. Les paramètres pour ces deux algorithmes sont choisis de manière empirique et opérationnelle (en particulier, on n'utilise pour la prévision à l'instant t que les données réellement disponibles avant l'instant t), suivant en cela la méthodologie de Devaine et al. [2013].

Sur la majorité des propriétés, les prévisions proposées sont proches (pour EWA et Ridge) et même meilleures (LASSO) que la meilleure simulation proposée par l'IFPEN.

Faisceaux de prévision

Les problèmes de calage d'historique sont généralement des problèmes mal posés. En effet, ils sont sur-paramétrés : même en imposant une cohérence géologique dans les paramètres des différents blocs de la grille, il reste encore beaucoup plus de modèles de sous-sol possibles que de mesures de données dynamiques, d'où une non-injectivité de l'application "paramètres du sous-sol $\mapsto$ prévisions du modèle géophysique". Par conséquent, les travaux récents ont tendance à considérer une variabilité sur les modèles de sous-sol considérés, et partant, une variabilité sur leurs observations futures.

Dans la Section 6.5 du chapitre 6, on adopte cet état d'esprit en fournissant des faisceaux de prévision (c'est-à-dire des ensembles de prévisions, qui dans notre cas seront des intervalles) court-terme et long-terme. Le fait de passer de prévision mensuelles à des prévisions multiples et surtout plus éloignées dans le temps augmente a priori l'incertitude, ce qui fait un second motif pour prévoir des faisceaux plutôt que des valeurs uniques.

On applique la méthodologie générale présentée au chapitre 5, afin d'obtenir des faisceaux de prévision portant sur les dernières années de l'étude, à partir des algorithmes Ridge et EWA. On effectue certaines adaptations liées au problème étudié, en particulier vis-à-vis du bruit et d'éventuels biais ; le problème du choix des paramètres est également présenté, mais non entièrement résolu (on propose néanmoins deux choix pour cette calibration dans le cas de l'algorithme Ridge).

Un critère de mesure de performance (l'efficacité) est proposé, ainsi qu'un "benchmark" (un point de référence), ce qui permet d'évaluer les performances des faisceaux proposés par Ridge (voir Section 1.5.2), avec souvent des résultats satisfaisants.

1.6.4. Pistes de recherche futures

Une piste intéressante serait de tenter de réaliser l'objectif habituel du calage d'historique tel que présenté plus haut (c'est-à-dire une rétro-ingénierie des paramètres du modèle de sous-sol).

Alors que les travaux présentés traitent séparément les différentes propriétés et les différents puits, une première étape (qui constitue déjà en elle-même un problème de recherche) dans ce sens serait de tenter une approche globale sur l'ensemble des propriétés, aboutissant à des résultats cohérents. Par exemple, on pourrait chercher des poids permettant de prévoir correctement l'ensemble des propriétés. Or ces propriétés, et les pertes associées, sont liées à des quantités physiquement différentes, avec des ordres de grandeur inégaux (et dépendant des unités utilisées). Cela rend difficile la comparaison des pertes entre ces différentes propriétés ; par conséquent être capable de gérer de manière "équitable" ces pertes liées à des quantités physiquement différentes dans la détermination des poids d'agrégation sera un point-clé de cette approche (il ne semble par exemple pas pertinent de sommer directement ces pertes issues de grandeurs physiques différentes). Arriver à concevoir une mise à l'échelle pertinente et automatique des différentes propriétés étudiées, et parvenir à des garanties théoriques intéressantes à ce sujet dans un contexte de suites individuelles, est à notre connaissance un problème ouvert.

Au contraire, on pourrait aussi se concentrer uniquement sur les capacités prédictives des

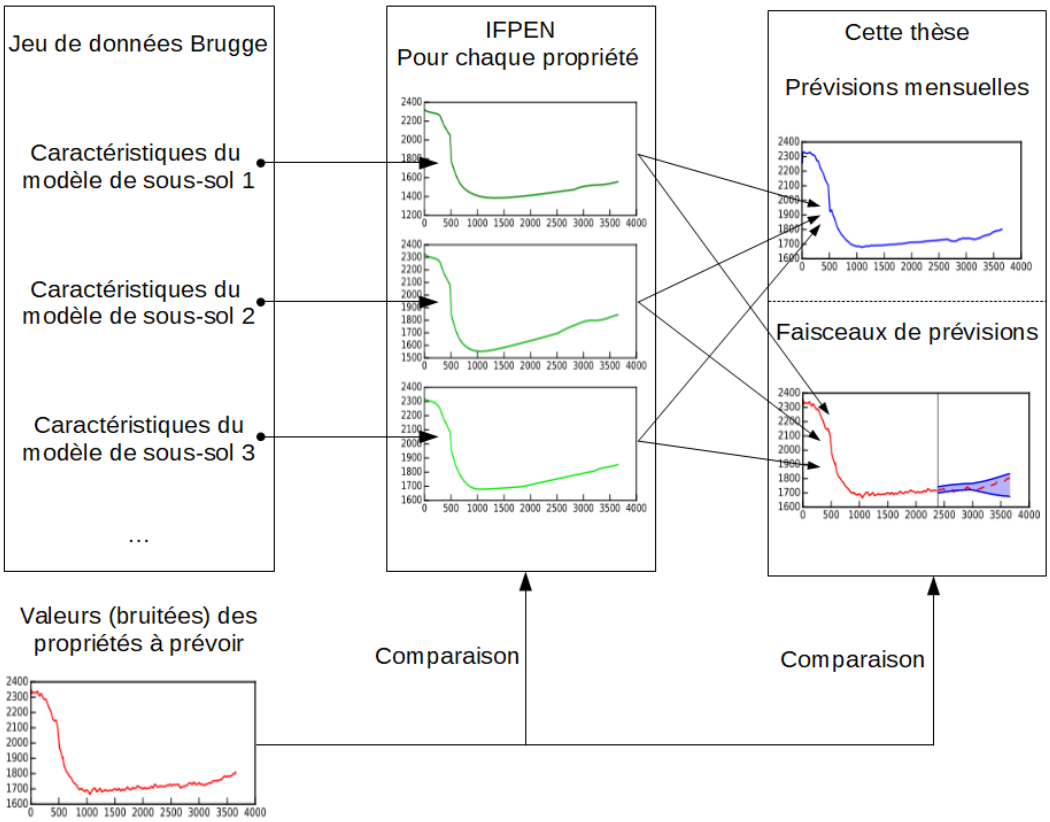


Figure 1.5: Plan d'étude du jeu de données Brugge

méthodes et modèles, et par exemple ajouter des experts utiles pour la prédiction mais n'ayant pas forcément de sens physique (experts sur-estimant ou sous-estimant systématiquement pour débiaiser plus facilement, expert prédisant comme l'instant précédent...).

Bien évidemment, il serait également intéressant de tester les méthodologies présentées (en particulier, vu leur nouveauté, celles liées aux faisceaux de prévision) sur d'autres jeux de données pour juger de leur efficacité dans différents contextes.

Chapter 2

Mathematical introduction

This mathematical introduction describes the frameworks in which this thesis works take place, and presents mathematical tools used in the following chapters, for self-containment.

First, it introduces the setting of forecasting with expert advice, and two corresponding important frameworks: the "individual sequences" setting and the "batch" setting. Then, it presents several classical algorithms for sequential aggregation, and some bounds they guarantee. Afterwards, it tackles a generalization of sequential aggregation: online convex optimisation, with a focus on the importance of the losses at hand, and introduces "online-to-batch" framework conversions. The final part is devoted to a discussion about regularization, which will be at the heart of Chapter 3.

2.1. Aggregation for individual sequences and in the batch setting

2.1.1. Aggregation for individual sequences

Online prediction. The framework of online forecasting consists in predicting a time-evolving quantity y_t in a convex space E (often $\mathbb{R}$ or $\mathbb{R}^d$), the "observation", at several instants ("rounds of prediction") $t = 1, 2, \ldots$ The total number of rounds, T, can be known beforehand or not. Before each round, the statistician predicts a value $\widehat{y}_t$, then the environment reveals the real observation value y_t, and the statistician suffers a loss $\ell(\widehat{y}_t, y_t)$. The goal of the statistician is to minimize the cumulative loss:

$$\widehat{L}_t := \sum_{s=1}^{t} \ell(\widehat{y}_s, y_s).$$

Classical loss functions include:

- the square loss: $\ell(\widehat{y}_t, y_t) = (\widehat{y}_t - y_t)^2$
- the absolute loss: $\ell(\widehat{y}_t, y_t) = |\widehat{y}_t - y_t|$
- the "0/1 loss": $\ell(\widehat{y}_t, y_t) = 1_{\widehat{y}_t \neq y_t}$

Online prediction is the framework of a large number of problems, from weather forecasting (Mauricette et al. [2009],Zhu and Pi [2014]) to traffic and travel times (Bajwa et al. [2005]), electricity consumption (Devaine et al. [2013]) or stock markets (Gokcan [2000]).

The case of experts aggregation. The experts aggregation setting consists in building the forecast $\widehat{y}_t$ by a combination (in this thesis, it will be a linear combination) of predictions of exterior forecasters, often called "experts". These can be algorithms, institutions, human beings... We assume that the number of experts is finite and denote it by K. The experts are indexed by $k \in \{1..K\}$, and release a forecast at the beginning of each round t. We denote by $f_{k,t}$ the forecast of the k-th expert for round t.

The forecast of an online aggregation algorithm at instant t is of the form:

$$\widehat{y}_t = \sum_{k=1}^{K} u_{k,t} f_{k,t}.$$

The object of an aggregation algorithm is to choose the weights $u_{k,t}$.

The process of online prediction with expert advice is summarized in Setting 3.

Some modified versions of this framework exist: for example, Blum [1997] and Freund et al. [1997] deal with specialized experts, that do not all provide forecasts at every round, and Gofer et al. [2013] tackles the problem of so-called "branching experts".

One can notice that this framework covers, for example, the case of geometric (and more generally multiplicative) averages, since applying the log function transforms these averages into "additive" averages considered in our framework.

Setting 3 Online forecasting with expert advice

Input: Decision space E, K experts.
for $t = 1, 2, \ldots$

1. Each expert k provides a forecast $f_{k,t}$
2. Based on all available data, the statistician chooses weights $(u_{1,t}, \ldots, u_{K,t})$ and a forecast $\widehat{y}_t = \sum_{k=1}^{K} u_{k,t} f_{k,t}$
3. The environment reveals a value y_t
4. The statistician incurs the loss $\ell(\widehat{y}_t, y_t)$ and the experts incur losses $\ell(f_{k,t}, y_t)$

Formally, expert aggregation is a particular case of linear regression, though with some specificities.

Firstly, if some (or even most) of the experts are assumed to perform well, then it makes sense to use convex algorithms (i.e., algorithm with nonnegative weights summing up to 1; we will then denote these weights by $p_{k,t}$ instead of $u_{k,t}$), whereas these algorithms may have less interest in more general linear regression setups. Yet, classical linear aggregation methods can also be used in the experts setting (cf. the Ridge and LASSO forecasters discussed in Section 2.2.3, and applied for instance in Chapter 6).

Secondly, the performance is generally not measured in some absolute manner, (e.g., through the cumulative loss):

$$\widehat{L}_t := \sum_{s=1}^{t} \ell(\widehat{y}_s, y_s)$$

but rather relatively to a reference benchmark linked to the experts. The cumulative loss of the k-th expert until round t will be denoted by:

$$L_{k,t} = \sum_{s=1}^{t} \ell(f_{k,s}, y_s).$$

In this thesis, we will often focus on the following quantity:

$$\widehat{R}_t := \sum_{s=1}^{t} \ell(\widehat{y}_s, y_s) - \min_{k=1..K} \sum_{s=1}^{t} \ell(f_{k,s}, y_s) := \widehat{L}_t - \min_{k=1..K} L_{k,t}$$

It is called the (cumulative) regret, because it measures how much the statistician would have improved his or her performance if he or she had followed the best (in hindsight) expert (supposing that the real observations would have remained $y_1, \ldots, y_t$).

It is also possible to compare the algorithm to other benchmarks that the best expert, for example the best constant combination of experts:

$$\sum_{s=1}^{t} \ell(\widehat{y}_s, y_s) - \min_{(\gamma_1, \ldots, \gamma_K) \in \Gamma} \sum_{s=1}^{t} \ell\left(\sum_{k=1}^{K} \gamma_k f_{k,s}, y_s \right)$$

with Γ being $\mathbb{R}^K$ (best linear combination), or the probabilistic simplex (best convex combination):

$$\Gamma = \left\{ (\gamma_1, \dots, \gamma_K) : \forall\ k,\ 0 \leqslant \gamma_k \leqslant 1 \text{ and } \sum_{k=1}^{K} \gamma_k = 1 \right\}.$$

This leads to various notions of regret.

Inequalities on the regret are sometimes described as "oracle inequalities" because they involve benchmarks (e.g., the minimum cumulative loss among the experts or their combinations) that depend on quantities that are not available to the learner during the forecasting period (the "best" expert and the "best" expert combination are only known in hindsight).

Individual sequences In the "individual sequences" setting, the data is assumed generated in an unknown deterministic way: no modeling (except boundedness) is performed on the data-generating process. It can take any value (within some range) at any instant. Therefore, theoretical results do not focus on the prediction of one precise round, rather they are often uniform bounds on the (cumulative) regret, valid for any individual sequence of values for the data (hence the name "individual sequences"):

$$\forall\ y_1, \dots, y_T, \quad \widehat{R}_T \leqslant r_T.$$

The bound r_T on the regret depends on the algorithm and on the hypotheses on the loss. For a loss bounded by M, a trivial bound on the regret for any algorithm is the linear (in T) quantity MT. Therefore, theoretical results aim at bounds that sublinear in T as for the regret:

$$r_T = o(T)$$

Equivalently, they aim at a vanishing bound on the averaged regret:

$$\widehat{R}_T/T \leqslant r_T/T \quad \text{with} \quad \lim_{T \to +\infty} r_T/T = 0.$$

The absence of major hypotheses prevents deducing links among the observations, and therefore limits the theoretical impact of a learning sample (preliminary sample given to the learner to help building the algorithm), which is usually not considered at all in the theorems. However, such a learning sample is often very useful in practical applications, allowing to "tune" the algorithm beforehand and avoid some chaotic first forecasts.

The absence of modeling and the uniform guarantees make these methods quite robust, that is why the term "robust aggregation" can be used to refer to the algorithms of expert aggregation in this individual sequences framework.

An important reference in this field is the monograph by Cesa-Bianchi and Lugosi [2006]. One can also cite the papers by Freund and Schapire [1997], Littlestone and Warmuth [1994], or Cesa-Bianchi et al. [2007].

2.1.2. Batch setting

In the so-called "batch setting", the learner has immediately access to all the past observations and forecasts (the full "batch" of them, hence the name), and generally has to make only one forecast (or several forecasts, but without supplementary information or feedback).

Formally, the learner has observed N vectors $x_1, \dots, x_N$ (the regressors), the corresponding N output values $y_1, \dots, y_N$ and tries to forecast the output value y_{N+1} given x_{N+1}. The $(x_i, y_i)_{i=1\dots N}$ form the learning sample. Contrary to the previous setup (individual sequences online learning), in the batch setting the learning sample plays a crucial role, as it is the main source of information for the "one-shot" forecast.

The process of batch prediction is summarized in Setting 4.

Setting 4 Batch forecasting process

Input: Decision space E.

Learning sample:

The statistician has access to N vectors $x_1, \dots, x_N$ and N corresponding values $y_1, \dots, y_N$

Forecasting stage:

1. A vector x_{N+1} is revealed
2. The statistician chooses a forecast $\widehat{y}$
3. The environment reveals a value y_{N+1}
4. The statistician suffers the loss $\ell(\widehat{y}, y_{N+1})$

In this context, the data is generally considered as random variables, upon which some stochastic assumptions are made (it is then called "stochastic batch setting"). In particular, to enable learning, the distribution of (x_{N+1}, y_{N+1}) is generally linked to the distributions of the $(x_i, y_i)_{i \leqslant N}$, for example through an assumption of stationarity, or even more strongly through an i.i.d. (independent and identically distributed) assumption.

The performance of an algorithm is measured by the risk, which is the expected loss:

$$\widehat{R}_{\widehat{y}} := \mathbb{E}\big[\ell(\widehat{y}(x_{N+1}), y_{N+1}) \,|\, (x_1, y_1, \dots, x_N, y_N)\big]$$

In the same spirit as above, one usually seeks guarantees not directly on the risk, but relatively to the risk of some benchmark, either in expectation or with high probability with respect to $(x_1, y_1), \dots, (x_N, y_N)$.

For instance, one can seek guarantees relatively to the best linear function with high probability:

$$\mathbb{P}_{(x_1,y_1),\dots,(x_N,y_N)}\left[\widehat{R}_{\widehat{y}} - \min_{f \text{ linear}} \mathbb{E}\big[\ell(f(x_{N+1}), y_{N+1})\big] \leqslant \zeta_N\right] \geqslant 1 - \delta_N \tag{2.1.1}$$

for some ζ_N and δ_N, where $\mathbb{P}_{(x_1,y_1),\dots,(x_N,y_N)}$ denotes the probability with respect to $(x_1, y_1), \dots, (x_N, y_N)$.

Two main settings exist: the case where both the x_i's and the y_i's are random variables is called "random design"; the case where the x_i's are fixed (chosen or not by the learner) and only the y_i's are random variables is called "fixed design".

One can notice that "classical" linear regression problems, such as finding an estimator $\widehat{\beta}$ of a vector β, given a matrix $\mathbb{X}$ and the values

$$Y = \mathbb{X}\beta + \varepsilon$$

where ε is a random noise vector, correspond to the batch setting. Similarly to the individual sequences forecasting, the term "aggregation" is usually used for situations where some individual predictors (the columns of $\mathbb{X}$ in the previous example) are assumed to be good and are used as benchmarks to be competed against. However, one should note that Tsybakov [2003] uses the term of "aggregation" for different problems, depending on the benchmark (which is the "best" vector in a set Γ) one wants to compete against. These problems are

- the model selection aggregation (Γ is the canonical basis of $\mathbb{R}^d$);
- the convex aggregation (Γ is the probabilistic simplex of $\mathbb{R}^d$);
- the linear aggregation ($\Gamma = \mathbb{R}^d$).

In the batch setting one can derive results on the forecasts (cf. (2.1.1)), like in the individual sequences framework, but also on the "quality" of the learning (e.g. bounds on $\|\widehat{\beta} - \beta\|$ in the previous example). These two objectives are known as the "prediction problem" and the "estimation problem".

2.1.3. Some comparisons between batch and individual sequences settings

One can see that in the batch setting, the N rounds taken into consideration are learning rounds, on which no predictions and no errors are made by the statistician, and which help to better know the data, so that the expected error at the final forecasting round tends to decrease when N grows. On the contrary, in the online setting, the T rounds taken into account are rounds of prediction, so the cumulative error tends to grow when T grows. However, the averaged regret $\widehat{R}_T/T$ tends (for "reasonable" algorithms!) to decrease; some links between averaging in online learning and batch results will be shown in the "online-to-batch" parts (Section 2.3.3 and Section 4.3). More generally, it is possible to "transfer" methods that are efficient in the online setting to the batch setting.

Conversely, we will make use in Chapters 5 and 6 of two algorithms, the Ridge and LASSO forecasters, that were initially developed for linear regression with fixed design, but that give interesting results for individual sequences predictions (cf. Section 2.2.3).

2.2. Algorithms for the forecasting of individual sequences

2.2.1. Introductory remarks

Failure of some naive algorithms. One of the simplest aggregation algorithm is the (arithmetic) average of the experts' predictions: $\widehat{y}_t = \sum_{k=1}^{K} f_{k,t}/K$. It is obvious that for most losses it can fail: even if some experts are good, some poorly-predicting experts may drive the average to suffer a large loss.

Another "natural" approach is to "follow the leader": allocate a weight 1 to the expert that has suffered the smallest cumulative loss until the instant to forecast (the "leader"), and a weight 0 to the other experts. This approach also fails: it suffices to consider a situation where, at each instant t, the "leader" until $t-1$ suffers the highest loss at time t, and then loses its leadership; such situations exist (in particular when two experts have a similar overall performance).

Necessity of hypotheses on the loss. Without hypotheses on the loss, it is not possible to guarantee a sublinear regret, whatever the algorithm. For instance, consider the loss $\ell(x,y) = 1_{x \neq y}$, the set of predictions $\{0,1\}$ and two constant experts: f_0 always forecasting 0 and f_1 always forecasting 1. With this loss, for any sequence of predictions, there exist a sequence of observations in $\{0,1\}$ that leads to a loss of 1 at each round ($y_t = 1$ when $\widehat{y}_t = 0$, and $y_t = 0$ when $\widehat{y}_t \neq 0$), so the cumulative loss of this sequence of predictions is T. As for the experts f_0 and f_1, at each round one of them receives a loss 0 and the other one receives a loss 1, so over T rounds one of them has a cumulative loss inferior (or equal) to $T/2$. As a consequence the regret is superior (or equal) to $T/2$, which is linear in T.

We will see below that a sufficient assumption to get interesting results is the convexity of the losses (in their first argument).

In the following, we will only consider convex losses.

2.2.2. Two convex algorithms

We first focus on two "convex" algorithms, in the sense that they output nonnegative weights summing up to 1, which we denote by $p_{k,t}$ instead of our notation $u_{k,t}$ for general weights:

$$\forall\ t,\ \forall\ k, \quad p_{k,t} \geqslant 0 \ \text{ and } \ \sum_{k=1}^{K} p_{k,t} = 1.$$

In particular, at each round, forecasts of convex algorithms are bounded between the uppest and lowest forecasts of the experts.

The Exponentially Weighted Average forecaster ("EWA")

General presentation. The Exponentially Weighted Average forecaster (abbreviated in "EWA" in the following) is a convex algorithm with positive weights, relying on an exponential decrease of the weights with respect to the cumulative losses, and on a parameter (called "learning rate") chosen by the statistician.

The weights provided by EWA, with fixed learning rate $\eta > 0$, are as follows: $p_{k,1} = 1/K$, and for each round $t \geqslant 2$,

$$p_{k,t} = \frac{\exp(-\eta L_{k,t-1})}{\sum_{i=1}^{K} \exp(-\eta L_{i,t-1})} \tag{2.2.1}$$

A higher learning rate η leads to a faster learning (larger differences among the weights), a lower η leads to a weight vector closer to the uniform vector (which corresponds to the "limit" case $\eta = 0$).

One can see that the weights output by EWA can be computed by a recursive update:

$$p_{k,t} = \frac{p_{k,t-1} \exp\Big(-\eta\, \ell(f_{k,t-1}, y_{t-1})\Big)}{\sum_{i=1}^{K} p_{i,t-1} \exp\Big(-\eta\, \ell(f_{i,t-1}, y_{t-1})\Big)}$$

First theoretical bounds. The next theorem and its corollary show that EWA, with a correctly tuned η, can achieve a regret of order $\sqrt{T}$.

Theorem 2.1. *If all the losses of the experts lie in the $[m, M]$ interval (where $m \leqslant M$ are real numbers) then the EWA algorithm with fixed parameter η incurs a regret bounded by:*

$$\widehat{R}_T := \widehat{L}_T - \min_{k=1,\dots,K} L_{k,T} \leqslant \frac{\log(K)}{\eta} + \eta \frac{(M-m)^2 T}{8} \tag{2.2.2}$$

Proof. It is a classical proof (see for instance Cesa-Bianchi and Lugosi [2006]). Denote by $w_{i,t} = \exp(-\eta L_{i,t})$ any "weight before normalization", and by $W_t := \sum_{i=1}^{K} w_{i,t}$ the "total weight before normalization". The proof relies on upper and lower bounds on the quantity $\log\left(\frac{W_T}{W_0}\right)$.

First, as a sum of positive terms is greater than the maximum of its terms, one has:

$$\begin{aligned}
\log\left(\frac{W_T}{W_0}\right) &= \log\left(\sum_{i=1}^{K} \exp(-\eta L_{i,T})\right) - \log(K) \\
&\geqslant \log\left(\max_{i=1,\dots,K} \exp(-\eta L_{i,T})\right) - \log(K) \\
&= -\eta \min_{i=1,\dots,K} L_{i,T} - \log(K).
\end{aligned} \tag{2.2.3}$$

As for the upper bound we use the following lemma, due to Hoeffding.

Lemma 2.2. *Let X be a random variable such that $a \leqslant X \leqslant b$. Then for any $s \in \mathbb{R}$,*

$$\log\big(\mathbb{E}[\exp(sX)]\big) \leqslant s\mathbb{E}[X] + \frac{s^2(b-a)^2}{8}$$

We apply this lemma to a random variable X such that $\mathbb{P}\Big(X = \ell(f_{i,t}, y_t)\Big) = w_{i,t-1} / \sum_{k=1}^{K} w_{k,t-1}$. Therefore:

$$\begin{aligned}
\log\left(\frac{W_t}{W_{t-1}}\right) &= \log\left(\frac{\sum_{i=1}^{K} w_{i,t-1} \exp(-\eta\, \ell(f_{i,t}, y_t))}{\sum_{k=1}^{K} w_{k,t-1}}\right) \\
&\leqslant -\eta \left(\frac{\sum_{i=1}^{K} w_{i,t-1} \ell(f_{i,t}, y_t)}{\sum_{k=1}^{K} w_{k,t-1}}\right) + \frac{\eta^2 (M-m)^2}{8} \\
&\leqslant -\eta\, \ell\left(\frac{\sum_{i=1}^{K} w_{i,t-1} f_{i,t}}{\sum_{k=1}^{K} w_{k,t-1}}, y_t\right) + \frac{\eta^2 (M-m)^2}{8} \\
&= -\eta\, \ell(\widehat{y}_t, y_t) + \frac{\eta^2 (M-m)^2}{8}
\end{aligned}$$

The last inequality comes from the convexity of ℓ. A telescopic sum over t leads to:

$$\log\left(\frac{W_T}{W_0}\right) \leqslant -\eta \widehat{L}_T + T\frac{\eta^2(M-m)^2}{8}. \tag{2.2.4}$$

Combining (2.2.3) and (2.2.4) and dividing by η leads to the result. ∎

If one has access to T, m and M, it is then possible to minimize the right-hand side of (2.2.2), by choosing $\eta = \sqrt{8\log(K)/((M-m)^2T)}$, which guarantees the following bound.

Corollary 2.3. *Under the hypothesis of Theorem 2.1, the EWA algorithm with parameter* $\eta = \sqrt{8\log(K)/((M-m)^2T)}$ *satisfies:*

$$\widehat{R}_T \leqslant (M-m)\sqrt{\frac{T\log(K)}{2}}$$

Tuning of the learning parameter. In many situations one does not have access to T or M. It is then useful to have an adaptive tuning of η, changing over time:
$p_{k,t} = \exp(-\eta_t L_{k,t-1}) / \sum_{i=1}^K \exp(-\eta_t L_{i,t-1})$, where η_t is set based on past data.

When m and M are known in advance, but not T, a possible approach is the so-called "doubling trick". It consists in dividing the rounds in "epochs", with epoch j containing the rounds $\{2^j, \dots, 2^{j+1}-1\}$. At the beginning of each epoch, the weights are reset, and within epoch j, whose length is 2^j, the EWA algorithm is run with learning parameter $\eta_t = \sqrt{8\log(K)/((M-m)^2 2^j)}$. Summing the regrets of all epochs give the following result.

Corollary 2.4. *The EWA algorithm run using the "doubling trick" guarantees that:*

$$\widehat{R}_T \leqslant (M-m)\frac{\sqrt{2}}{\sqrt{2}-1}\sqrt{\frac{T\log(K)}{2}}$$

Proof. The index of the last epoch is $J = \lfloor \log_2(T) \rfloor$. Therefore, the sum of the regrets over all the epochs is bounded by:

$$\sum_{j=0}^{J}(M-m)\sqrt{\frac{2^j\log(K)}{2}} = \frac{\sqrt{2}^{J+1}-1}{\sqrt{2}-1}(M-m)\sqrt{\frac{\log(K)}{2}} \leqslant (M-m)\frac{\sqrt{2}}{\sqrt{2}-1}\sqrt{\frac{T\log(K)}{2}}.$$

∎

If m and M are also unknown, one can apply variants that use a parameter η_t independent of T, m and M. A possibility is provided in De Rooij et al. [2014].

Proposition 2.5. *If one defines* $\overline{\ell_t} := \sum_k \ell(f_{k,t}, y_t)/K$ *and:*

$$\delta_t = \frac{1}{\eta_t}\sum_{k=1}^{K} p_{k,t}\exp\left(-\eta_t\left(\ell(f_{k,t},y_t) - \overline{\ell_t}\right)\right)$$

then the EWA algorithm run with learning parameter $\eta_t = \log(K)/\sum_{s=1}^{t-1}\delta_s$ *guarantees:*

$$\widehat{R}_T \leqslant (M-m)\sqrt{T\log(K)} + (M-m)\left(2 + \frac{4}{3}\log(K)\right).$$

However, all these choices for η or η_t, tuned to tackle worst-case scenarios, tend to be too conservative to get top performance with real data sets. In particular, the η_t are often too small. One can rather use more "data-driven" choices for η_t. In Chapter 6, we will use at round t the empirically best parameter for the rounds until $t-1$, i.e., the parameter η that would have led to the smallest cumulative loss on rounds until $t-1$ if one had run the EWA algorithm with this fixed parameter η. This choice is not associated with theoretical guarantees, but it often leads to a better practical performance, since it is more adapted to the data.

The Fixed-Share EWA forecaster

The previous algorithm EWA is based on the cumulative loss of each expert. It is therefore not carved to take advantage of changes in the performance of the experts (in particular, experts incurring a large loss during the first rounds are bound to have small weights for a long time). The following modified version of EWA, introduced by Herbster and Warmuth [1998], allows to benefit from recent improvements of any expert. It is called "Fixed-Share EWA" because it guarantees a minimal weight (the "fixed share") for each expert.

The weights $p_{k,t}$ of Fixed-Share EWA, with fixed parameters α (share) and η (learning rate) are given as follows: $p_{k,1} = \frac{1}{K}$, and for each round $t \geqslant 2$:

$$p_{k,t} = \frac{\alpha}{K} + (1-\alpha)\frac{p_{k,t-1}\exp\Big(-\eta\,\ell(f_{k,t-1}, y_{t-1})\Big)}{\sum_{i=1}^{K} p_{i,t-1}\exp\Big(-\eta\,\ell(f_{i,t-1}, y_{t-1})\Big)} \tag{2.2.5}$$

See Herbster and Warmuth [1998] for a "shifting bound" guaranteed by Fixed-Share EWA, i.e., a bound with respect to the minimal loss one can get by forecasting the same values that a precise expert, but with the possibility of shifting from one expert to another one a defined number of times.

2.2.3. Two non-convex regularized algorithms

We now focus on two algorithms that do not output convex weights but general linear weights (signed weights whose sum does not need to equal 1). They are "regularized" algorithms, i.e. the weights chosen for round t are minimizers of the quantity:

$$\sum_{s=1}^{t-1} \ell\left(\sum_{k=1}^{K} u_k f_{k,s}, y_s\right) + \Psi(u_1, \ldots, u_K)$$

with Ψ depending on the algorithm.

This notion of regularization will be at the heart of Section 2.4, where we will also explain its interest in terms of avoiding overfitting issues.

The Ridge regression forecaster

The Ridge regression forecaster is a regularized "least squares" estimator, introduced by Hoerl [1962]. It relies on a regularization parameter λ, and on a Euclidean regularization $\Psi : u \mapsto \|u\|_2^2$.

Consider some instant t, and denote:

- by f_t the vector of all the experts' forecasts at time t: $f_t = (f_{1,t}, \ldots, f_{K,t})^\top$,
- by $F_t = (f_{s,k})_{\substack{s \leqslant t \\ k \leqslant K}}$ the $t \times K$ matrix of the experts' forecasts up to time t (notice that we inverse the order of indices of the experts' forecasts),
- by $Y_t = (y_1, \ldots, y_t)^T$ the column-vector of the observations up to time t ($.^T$ meaning "transpose").

The weight vector output by Ridge $u_{t+1}^R := \begin{pmatrix} u_{1,t+1}^R \\ \ldots \\ u_{K,t+1}^R \end{pmatrix}$ is the minimizer of:

$$u = \begin{pmatrix} u_{1,t+1} \\ \ldots \\ u_{K,t+1} \end{pmatrix} \longmapsto \|Y_t - F_t u\|_2^2 + \lambda \|u\|_2^2. \tag{2.2.6}$$

The weight vector generated by Ridge and therefore the Ridge regression forecasts, are linear mappings of the observations, as is shown in Lemma 2.6.

Lemma 2.6. *The weights and forecasts provided by the Ridge algorithm (with regularization parameter λ) are linear with respect to the observations vector Y_t:*

$$\begin{pmatrix} u_{1,t+1}^R \\ \ldots \\ u_{K,t+1}^R \end{pmatrix} = M_t Y_t \qquad \text{with } M_t := (F_t^T F_t + \lambda I_K)^{-1} F_t^T$$

thus

$$\widehat{y}_{t+1} = V^{t+1} Y_t \qquad \text{with } V^{t+1} := (f_{1,t+1}, \ldots, f_{K,t+1})(F_t^T F_t + \lambda I_K)^{-1} F_t^T \in \mathbb{R}^t$$

Proof. The weight vector u_{t+1}^R given by the Ridge regression forecaster is defined in (2.2.6) as the minimizer of a smooth convex function. Therefore, it is a zero of its gradient:

$$-2F_t^T \left(Y_t - F_t u_{t+1}^R\right) + 2\lambda u_{t+1}^R = 0.$$

This leads directly to $\left(F_t^T F_t + \lambda I_K\right) u_{t+1}^R = F_t^T Y_t$.

For any $\lambda > 0$, the matrix $F_t^T F_t + \lambda I_K$ is invertible (symmetric definite positive), so one gets the first result:

$$u_{t+1}^R = (F_t^T F_t + \lambda I_K)^{-1} F_t^T Y_t.$$

These weights lead to the forecast (at time $t+1$):

$$\widehat{y}_{t+1} = f_{t+1}^\top u_{t+1}^R = V^{t+1} Y_t.$$

■

One has the following bound on the regret of the Ridge forecaster. We will use for convenience the notation: $\ell_t(u) := \left(\sum_{k=1}^K u_k f_{k,t} - y_t\right)^2$.

Theorem 2.7. *Assume that the $|f_{k,t}|$ are bounded by F, and that the y_t are bounded by Y. Then, for all $\lambda > 0$, the Ridge forecaster run with regularization parameter λ guarantees, for all $u = (u_1, \ldots, u_K)^\top \in \mathbb{R}^K$ that:*

$$\sum_{t=1}^T \ell_t(u_t^R) - \sum_{t=1}^T \ell_t(u) \leqslant \lambda \|u\|_2^2 + 4KY^2 \left(1 + \frac{KF^2T}{\lambda}\right) \log\left(1 + \frac{F^2T}{\lambda}\right) \tag{2.2.7}$$

Proof. The proof is derived from Azoury and Warmuth [2001] and Vovk [2001].

Firstly, an induction gives that:

$$\forall u \in \mathbb{R}^K, \quad \sum_{t=1}^T \ell_t(u_{t+1}^R) - \sum_{t=1}^T \ell_t(u) \leqslant \lambda \left(\|u\|_2^2 - \|u_2^R\|_2^2\right) \tag{2.2.8}$$

It is true for $T = 1$ by definition of u_2^R. If it is true for some $T \geqslant 1$, then by definition of u_{T+2}^R,

$$\begin{aligned}
\forall u \in \mathbb{R}^K, \ \sum_{t=1}^{T+1} \ell_t(u) + \lambda\|u\|_2^2 &\geqslant \sum_{t=1}^{T+1} \ell_t(u_{T+2}^R) + \lambda\|u_{T+2}^R\|_2^2 \\
&\geqslant \ell_{T+1}(u_{T+2}^R) + \sum_{t=1}^{T} \ell_t(u_{t+1}^R) + \lambda\|u_2^R\|_2^2 \\
&= \sum_{t=1}^{T+1} \ell_t(u_{t+1}^R) + \lambda\|u_2^R\|_2^2
\end{aligned}$$

so it is also true for $T + 1$.

Secondly, let us prove that:

$$\sum_{t=1}^T \left(\ell_t(u_t^R) - \ell_t(u_{t+1}^R)\right) \leqslant 2 \left(\sum_{t=1}^T f_t^\top \left(F_t^T F_t + \lambda I_K\right)^{-1} f_t\right) \times \max_{t \leqslant T} \ell_t(u_t^R) \tag{2.2.9}$$

Using the fact that $F_t^\top F_t = F_{t-1}^\top F_{t-1} + f_t f_t^\top$, one can easily derive from the proof of Lemma 2.6 that:

$$\begin{aligned}
u_{t+1}^R &= \left(F_t^\top F_t + \lambda I_K\right)^{-1} \left(F_{t-1}^\top Y_{t-1} + y_t f_t\right) \\
&= \left(F_t^\top F_t + \lambda I_K\right)^{-1} \left(\left(F_t^\top F_t + \lambda I_K - f_t f_t^\top\right) u_t^R + y_t f_t\right) \\
&= u_t^R - \left(F_t^\top F_t + \lambda I_K\right)^{-1} \left((f_t^\top u_t^R) - y_t\right) f_t
\end{aligned}$$

Combined with the gradient convexity inequality, this gives:

$$\begin{aligned}\ell_t(u_t^R) - \ell_t(u_{t+1}^R) &\leqslant \nabla \ell_t(u_t^R)^\top (u_t^R - u_{t+1}^R) \\ &= 2\left(f_t^\top u_t^R - y_t\right) f_t^\top \left(F_t^\top F_t + \lambda I_K\right)^{-1} \left(f_t^\top u_t^R - y_t\right) f_t \\ &= 2 f_t^\top \left(F_t^\top F_t + \lambda I_K\right)^{-1} f_t \times \ell_t(u_t^R)\end{aligned}$$

Summing over t gives Inequality (2.2.9).

Thirdly, one can use the proof of Lemma 11 of Hazan et al. [2007] (applied to $u_t = f_t$ and $\varepsilon = \lambda$) to get that:

$$\sum_{t=1}^{T} f_t^\top \left(F_t^\top F_t + \lambda I_K\right)^{-1} f_t \leqslant \log\left(\frac{\det\left(F_t^\top F_t + \lambda I_K\right)}{\det\left(\lambda I_K\right)}\right)$$

Let us denote by $\lambda_1, \cdots, \lambda_K$ the eigenvalues of the symmetric positive matrix $F_t^\top F_t$. Then $\det\left(F_t^\top F_t + \lambda I_K\right) = \prod_{k=1}^K (\lambda + \lambda_k)$ and one has that:

$$\sum_{t=1}^{T} f_t^\top \left(F_t^\top F_t + \lambda I_K\right)^{-1} f_t \leqslant \sum_{k=1}^{K} \log\left(1 + \frac{\lambda_k}{\lambda}\right)$$

Since the $f_{k,t}$ are bounded by F, one has that $\sum_{k=1}^K \lambda_k = \mathrm{Tr}(F_t^\top F_t) \leqslant KTF^2$.

Therefore, as $x \mapsto \log(1 + x/\lambda)$ is concave, one has:

$$\frac{1}{K}\sum_{k=1}^{K} \log\left(1 + \frac{\lambda_k}{\lambda}\right) \leqslant \log\left(1 + \frac{\sum_{k=1}^K \lambda_k}{K\lambda}\right) \leqslant \log\left(1 + \frac{TF^2}{\lambda}\right)$$

So one has:

$$\sum_{t=1}^{T} f_t^\top \left(F_t^\top F_t + \lambda I_K\right)^{-1} f_t \leqslant K \log\left(1 + \frac{TF^2}{\lambda}\right) \tag{2.2.10}$$

Fourthly, it remains to bound $\max_{t \leqslant T} \ell_t(u_t^R)$. Using the inequality $(a-b)^2 \leqslant 2(a^2+b^2)$, Cauchy-Schwarz inequality and the fact that $\|f_t\|_2^2 \leqslant KF^2$, one has that:

$$\ell_t(u_t^R) \leqslant 2\left((f_t^\top u_t^R)^2 + y_t^2\right) \leqslant 2\left(KF^2 \|u_t^R\|_2^2 + Y^2\right)$$

By definition of u_t^R, comparing it to the null vector gives, in terms of regularized cumulative losses:

$$\lambda \|u_t^R\|_2^2 + \sum_{s=1}^{t-1} \ell_s(u_t^R) \leqslant 0 + \sum_{s=1}^{t-1} (0 - y_s)^2 \leqslant (t-1)Y^2$$

Therefore, $\|u_t^R\|_2^2 \leqslant TY^2/\lambda$ and:

$$\max_{t \leqslant T} \ell_t(u_t^R) \leqslant 2Y^2\left(1 + \frac{KF^2T}{\lambda}\right) \tag{2.2.11}$$

Combining all the previous results, namely (2.2.8), (2.2.9), (2.2.10) and (2.2.11), concludes the proof. ■

The LASSO forecaster

The Least Absolute Shrinkage and Selection Operator (LASSO) forecaster is, like the Ridge regression forecaster, a regularized "least squares" estimator. It has been introduced by Tibshirani [1996]. It will be presented with more details in Section 2.4.3. With the same notations as above, the weight vector output by LASSO $u_{t+1}^L := (u_{1,t+1}^L, \dots, u_{K,t+1}^L)^\top$ is defined as a minimizer of:

$$u = \begin{pmatrix} u_{1,t+1} \\ \dots \\ u_{K,t+1} \end{pmatrix} \longmapsto \|Y_t - F_t u\|_2^2 + \lambda \|u\|_1, \tag{2.2.12}$$

for some parameter λ tuned by the statistician.

If one denotes $\lambda' := \lambda/N$, then it is clearly equivalent to define u_{t+1}^L as a minimizer of: $u \longmapsto \|Y_t - F_t u\|_2^2/N + \lambda' \|u\|_1$, as it will be done in Section 2.4.

There are no known non-trivial bounds for the LASSO forecaster in the context of individual sequences.

Let us get back to the notation $\ell_t(u)$ used in the proof of Theorem 2.7: it emphasizes the fact that the loss of an aggregation algorithm is not only linked to its forecast, but also to the weight vector used to output this forecast. Focusing on this weight vector and on its link with the loss is the spirit of the framework in the next section takes place.

2.3. Online convex optimization

2.3.1. Framework

In this section, and in Chapter 4, we will go beyond linear regression with convex losses and tackle a more general framework: online convex optimization. In this setup, we still provide at each step a vector w_t and get then a loss function ℓ_t that we want to minimize, but few assumptions will be made on the ℓ_t (typically convexity).

Loss in the regression setup:

$$w_t \to w_t^\top x_t \to (w_t^\top x_t, y_t) \to \ell_t(w_t^\top x_t, y_t)$$

(when typically ℓ_t will be a constant).

Loss in the online convex optimization setup:

$$w_t \to \ell_t(w_t)$$

Thus, in this setting, ℓ_t encompasses both the predictions x_t and the observation y_t. The domain of interest $\mathcal{K} \subset \mathbb{R}^d$ is convex; we will assume it is closed. It is generally fixed, but in some cases it can vary over time.

Similarly to Section 2.2, the number of rounds T may be known beforehand, or not (in Chapter 4 it will be known).

The ℓ_t are deterministic and sequentially picked over time by the environment: they can be any convex function. The goal is then to minimize the following regret:

$$\sum_{t=1}^{T} \ell_t(w_t) - \min_{u \in \mathcal{K}} \sum_{t=1}^{T} \ell_t(u)$$

.

We recall that the mere hypothesis of convexity (without any smoothness assumption) guarantees the existence of a non-empty subgradient in any point in the interior $\text{int}(\mathcal{K})$ of $\mathcal{K}$:

$$\forall\, x \in \text{int}(\mathcal{K}), \quad \partial \ell_t(x) \neq \emptyset, \quad \text{where} \quad \partial \ell_t(x) := \{v : \forall\, y \in \mathcal{K}, \ell_t(y) \geqslant \ell_t(x) + v^\top (y - x)\}.$$

Of course, other assumptions can be added on the losses ℓ_t: boundedness, (sub)gradient boundedness, hypotheses about curvature (e.g., exp-concavity, strong convexity). Their impacts are presented in the next section.

We will focus on the case of first-order information, where one has access after time t only to $\nabla \ell_t(w_t)$, and to $\ell_t(w_t)$, but not to the entire function ℓ_t. This framework is summarized in Setting 5.

Setting 5 Online convex optimization (with first-order information) framework

for $t = 1, 2, \ldots$

1. Play $w_t \in \mathcal{K}$
2. The environment picks a convex loss function $\ell_t : \mathcal{K} \to \mathbb{R}$
3. Increase cumulative loss by $\ell_t(w_t)$ and observe (sub)gradient $g_t \in \partial \ell_t(w_t)$ as well as $\ell_t(w_t)$

2.3.2. Different losses lead to different regrets –and require different algorithms?

General convex losses

It is clear that online convex optimization algorithms are likely to use ideas from different fields of mathematics: statistics, optimization, geometry, etc.

A standard algorithm in optimization (a fortiori, in convex optimization, which prevents local non-global minima), is gradient descent. Its online version has been introduced in Zinkevich [2003] (see also Cesa-Bianchi [1999]). It is based on the following update:

$$w_{t+1} = w_t - \eta_t \nabla \ell_t(w_t).$$

The choice of the learning rate η_t is crucial. In the case of bounded domain and bounded gradient (at any time and any point), online gradient descent with η_t of order $1/\sqrt{t}$ achieves the optimal regret for the framework described above (general convex functions): $O(\sqrt{T})$ (cf. Zinkevich [2003]; see also Hazan [2016]).

Yet, this $1/\sqrt{t}$ learning rate might not be optimal in practice. To overcome part of the problem of tuning η_t, Duchi et al. [2011] introduce a new algorithm, called Adagrad. It runs a separate learning rate for each dimension, and (still after one choice of the scaling factor η by the statistician) automatically tunes these learning rates based on the previous observed gradients. Denoting w_t^j and g_t^j the j–th component, respectively, of the output and of the gradient, the update is given by:

$$w_{t+1}^j = w_t^j - \frac{\eta}{\sqrt{\sum_{s=1}^t (g_s^j)^2}} g_t^j.$$

Even if it was created for convex problems, Adagrad has been successfully used for non-convex situations (cf. Gupta et al. [2014]).

If a regret of $\Omega(\sqrt{T})$ is unavoidable (cf. Hazan et al. [2007]) when dealing with general convex losses (which include linear functions, without curvature), supplementary assumptions on the loss allow improved bounds.

Strongly convex losses

If convexity lower bounds the function by a linear quantity (it is the definition of the subgradient), a stronger hypothesis is to lower bound the function by a quadratic quantity. This is called strong convexity. A function $f : \mathcal{K} \mapsto \mathbb{R}$ is λ-strongly convex if it is convex and:

$$\forall\ x, y \in \mathcal{K}, \quad \forall\ \nabla f(x) \in \partial f(x), \quad f(y) \geqslant f(x) + \nabla f(x)^\top (y - x) + \frac{\lambda}{2}\|y - x\|_2^2.$$

This strong curvature leads to improved regret guarantees: a regret of $O(\log(T))$ can be achieved with online gradient descent, with parameter η_t of order $1/t$ (cf Hazan et al. [2007]). It is natural to use a learning rate smaller than for general convex functions, since the extra curvature increases the changes in the function, even in small areas.

Exp-concave losses

An intermediate assumption is exp-concavity. A function f is α-exp-concave (with $\alpha > 0$) if $\exp(-\alpha f)$ is concave. One can easily see that it implies convexity, and that it is implied by strong convexity. The following lemma (proved for instance in Hazan et al. [2007]) illustrates that intermediate situation between convexity and strong convexity:

Lemma 2.8. *Let $f : \mathcal{K} \mapsto \mathbb{R}$ be an $\alpha-$exp-concave function. Let us assume that $\mathcal{K}$ is bounded with diameter D, and that for any point $x \in \mathcal{K}$, one has $\|\nabla f(x)\|_2 \leqslant G$. Then, for $\beta \leqslant \min\left(1/(4GD), \alpha\right)/2$, one has:*

$$\forall\ x, y \in \mathcal{K}, f(y) \geqslant f(x) + \nabla f(x)^\top (y - x) + \beta \left(\nabla f(x)^\top (y - x)\right)^2$$

Using Cauchy-Schwarz inequality (and the assumption $\|\nabla f(x)\|_2 \leqslant G$), the inequality of the previous lemma also holds if f is λ-strongly convex, taking $\beta = \lambda/(2G^2)$.

In the context of the first-order information framework, one does not have access to the second-order derivative of ℓ_t (if ever it exists). Therefore, one cannot resort to the second-order Taylor expansion:

$$f(y) \simeq f(x) + \nabla f(x)^\top (y-x) + \frac{1}{2}(y-x)^\top \nabla^2 f(x)(y-x)$$

where $\nabla^2 f(x)$ stands for the Hessian matrix of f at the point x. But Lemma 2.8 gives another way to be more precise than with mere linear approximation: adding a quadratic term involving the gradient, proportional to $\big(\nabla f(x)^\top (y-x)\big)^2$.

This idea of "replacing the Hessian matrix with a quadratic gradient term" leads to an interesting variation of Newton's method. The starting point is that a possibility for minimizing a convex function is to look at the zeros of its gradient. This can be done via Newton's method. The "Online Newton Step" algorithm, introduced in [Hazan et al., 2007], modifies this Newton's method, especially by replacing the Hessian matrix by $\varepsilon I_d + \sum_{s=1}^t g_t g_t^\top$ (with ε a term depending in particular on the parameter of "exp-concavity").

Under the assumption of exp-concavity for the losses, [Hazan et al., 2007] guarantees a bound $O(d \log(T))$ on the regret with the Online Newton Step method (cf. Lemma 4.8).

Adaptivity

We have introduced so far in this section several algorithms, with different tunings of their learning rates, each one particularly adapted to a specific category of losses. To be optimal, the tuning of their learning parameter often requires the knowledge of the degree of curvature (parameter of exp-concavity, etc.). A natural objective is then to build adaptive algorithms, i.e. algorithms that are able to be efficient on several categories of convex losses, without knowing the degree of curvature beforehand. The MetaGrad algorithm, presented in Chapter 4, is an important step in this direction.

2.3.3. From online setting results to batch setting results

General approach. It is possible to transfer performance bounds in the online setting to the batch setting, in particular for convex losses. A "classical" way of doing so is given in Theorem 2.9 below (cf. Littlestone [1989]). Here we leave the "first-order information" framework: we assume that we have access after round t to the whole loss ℓ_t. For instance, in the case of linear regression, we have access to the vector x_t and to the observation y_t, and therefore to the loss $w \mapsto \ell(w^\top x_t, y_t)$.

If the training sample is $(\ell_1, \dots, \ell_T)$, the general idea consists in treating this training sample, totally available from the start, as if it were discovered in an online fashion, applying an online algorithm and getting the sequence:

$$w_1, \quad w_2 = w_2(\ell_1), \quad w_3 = w_3(\ell_1, \ell_2), \quad \dots, \quad w_T = w_T(\ell_1, \dots, \ell_{T-1}).$$

Then, the forecast is built upon these intermediate forecasts: $w = w\,(w_1, \dots, w_{T-1})$. Twists can be added, using the actual availability of the data, as in Proposition 2.10, or allowing

intermediate restarts that might be suboptimal in terms of the cumulative loss (because here, the cumulative loss is of no direct importance) but are useful for the real one-shot prediction of interest, like in Chapter 4.

A classical online-to-batch conversion. The following theorem only relies on the online regret of algorithms, not on the way they build their forecasts, so it can be applied to aggregation algorithms but also to other kinds of methods.

Theorem 2.9. *Consider a set S of convex losses, a subset $\mathcal{K}'$ of $\mathcal{K}$, and an online algorithm such that, for any sequence of losses $(\bar{\ell}_1, \dots, \bar{\ell}_T) \in S^T$, the outputs $(w_1, \dots, w_T)$ satisfy:*

$$\sum_{s=1}^{T} \bar{\ell}_s(w_s) - \inf_{u \in \mathcal{K}'} \sum_{s=1}^{T} \bar{\ell}_s(u) \leqslant r_T \tag{2.3.1}$$

where r_T is some scalar value.

Consider an i.i.d. sequence of losses $\ell_1, \dots, \ell_{T+1}$ drawn in S, where the training sample $\ell_1, \dots, \ell_T$ is given and ℓ_{T+1} must be minimized. Then, applying in an online fashion this algorithm to the training sample, as described above, and forecasting as the average of the outputs:

$$w = \frac{1}{T} \sum_{s=1}^{T} w_s \left(\ell_1, \dots, \ell_{s-1}\right)$$

gives the guarantee:

$$\mathbb{E}\big[\ell_{T+1}(w)\big] - \inf_{u \in \mathcal{K}'} \mathbb{E}\big[\ell_{T+1}(u)\big] \leqslant \frac{r_T}{T}$$

Proof. We first use the convexity of the loss ℓ_{T+1} to write:

$$\mathbb{E}\left[\ell_{T+1}(w)\right] = \mathbb{E}\left[\ell_{T+1}\left(\frac{1}{T}\sum_{s=1}^{T} w_s\right)\right] \leqslant \frac{1}{T}\mathbb{E}\left[\sum_{s=1}^{T} \ell_{T+1}\left(w_s\right)\right] \tag{2.3.2}$$

Then, we use the fact that w_s is only based on $(\ell_1, \dots, \ell_{s-1})$ so it is independent of ℓ_s and ℓ_{T+1}, since the ℓ_t are i.i.d. Therefore, $\mathbb{E}\left[\ell_s\left(w_s\right)\right] = \mathbb{E}\left[\ell_{T+1}\left(w_s\right)\right]$. We can then use the "online guarantee hypothesis" (2.3.1), which is true for any individual sequence of losses, and therefore also in expectation:

$$\mathbb{E}\left[\sum_{s=1}^{T} \ell_s\left(w_s\right)\right] \leqslant \mathbb{E}\left[\inf_{u \in \mathcal{K}'} \sum_{s=1}^{T} \ell_s(u) + r_T\right]$$

It suffices then to write:

$$\mathbb{E}\left[\inf_{u \in \mathcal{K}'} \sum_{s=1}^{T} \ell_s(u)\right] \leqslant \inf_{u \in \mathcal{K}'} \mathbb{E}\left[\sum_{s=1}^{T} \ell_s(u)\right]$$

and to divide by T to get:

$$\mathbb{E}\left[\ell_{T+1}(w)\right] \leqslant \frac{1}{T} \inf_{u \in \mathcal{K}'} \mathbb{E}\left[\sum_{s=1}^{T} \ell_s(u)\right] + \frac{r_T}{T}$$

Using again the fact that the ℓ_s and ℓ_{T+1} are i.i.d., we have: $\mathbb{E}\left[\ell_s(u)\right] = \mathbb{E}\left[\ell_{T+1}(u)\right]$, therefore

$$\inf_{u\in\mathcal{K}'} \mathbb{E}\left[\sum_{s=1}^{T} \ell_s(u)\right] = T \inf_{u\in\mathcal{K}'} \mathbb{E}\left[\ell_{T+1}(u)\right],$$

which gives the desired result. ∎

As noticed by Audibert [2009], one can see that the construction of w does not use ℓ_T, so if the "online guarantee hypothesis" (2.3.1) remains true one step further (i.e. replacing T by $T+1$), then one can include the "last online forecast" w_{T+1} into w:

$$w = \frac{1}{T+1}\sum_{s=1}^{T+1} w_s\left(\ell_1, \ldots, \ell_{s-1}\right)$$

It guarantees a batch regret of at most $r_{T+1}/(T+1)$ instead of r_T/T.

A randomized online-to-batch conversion, useful for non-convex losses. It is possible to apply the ideas of the previous proof to the case of a non-convex loss, by replacing the averaging of the forecasts by a random choice among them. It is also possible to take into account the elements of the learning sample in a different order, for example a reverse order to prioritize the latest values. Those two ideas lead to the following lemma, seen in lectures slides of Bartlett [2011].

Proposition 2.10. *Consider the setting of Theorem 2.9, excepted that the ℓ_t are not necessarily convex. Then picking randomly and uniformly an integer J in $\{0, 1, \ldots, T-1\}$ and outputting $w = w_{T-J}\left(\ell_{J+1}, \ldots, \ell_{T-1}\right)$ gives the following guarantee:*

$$\mathbb{E}\left[\ell_{T+1}(w)\right] - \inf_{u\in\mathcal{K}'} \mathbb{E}\left[\ell_{T+1}(u)\right] \leqslant \frac{r_T}{T}$$

Remark: for $J = T-1$, the output is w_1, which is independent of the ℓ_k's.

Proof. First, we use the "tower rule" on the independent randomization to transfer the mean outside the loss:

$$\begin{aligned}
\mathbb{E}\big[\ell_{T+1}(w)\big] &= \mathbb{E}_J\Big[\mathbb{E}\big[\ell_{T+1}(w)\big|J\big]\Big] \\
&= \mathbb{E}_J\Big[\mathbb{E}\big[\ell_{T+1}\left(w_{T-J}\left(\ell_{J+1}, \ldots, \ell_{T-1}\right)\right)\big|J\big]\Big] \\
&= \frac{1}{T}\sum_{s=0}^{T-1} \mathbb{E}\left[\ell_{T+1}(w_{T-s}\left(\ell_{s+1}, \ldots, \ell_{T-1}\right))\right]
\end{aligned}$$

Then, since the ℓ_s are i.i.d., one can shift the variables into the expectation, and then reorganize the sum:

$$\begin{aligned}
\frac{1}{T}\sum_{s=0}^{T-1} \mathbb{E}\left[\ell_{T+1}\left(w_{T-s}\left(\ell_{s+1}, \ldots, \ell_{T-1}\right)\right)\right] &= \frac{1}{T}\sum_{s=0}^{T-1} \mathbb{E}\left[\ell_{T+1}\left(w_{T-s}\left(\ell_1, \ldots, \ell_{T-s-1}\right)\right)\right] \\
&= \frac{1}{T}\sum_{s=1}^{T} \mathbb{E}\left[\ell_{T+1}\left(w_s\left(\ell_1, \ldots, \ell_{s-1}\right)\right)\right]
\end{aligned}$$

As the last expression corresponds to the one in (2.3.2), the remaining part of the proof is similar to the one of Theorem 2.9. ■

One can thus see that the use of a "reverse order" in the procedure
–using $w = w_{T-J}(\ell_{J+1}, \dots, \ell_T)$ instead of $w_J(\ell_1, \dots, Z_{\ell-1})$– does not change the results.

One can also notice that, similarly to the previous case, if one has online guarantees up to $T+1$, one can slightly change the procedure to get a $r_{T+1}/(T+1)$ bound, instead of a r_T/T bound: it suffices to pick T in $\{0, 1, \dots, T\}$ and to output $w = w_{T+1-J}(\ell_{J+1}, \dots, \ell_T)$.

The slides of Bartlett [2011] also use more sophisticated techniques to tackle the case where the ℓ_t are stationary, but not i.i.d.

Being suboptimal in the online setting, and optimal in the batch setting. In an "online-to-batch conversion", the goal is to get a good predictor for the batch setting, not to control the cumulative error. This fact allows to use, in the process, parameters or starting points that would not give the best guarantees for an online cumulative error, but which will allow to secure enough information to get good guarantees for the batch forecast. It is the case of the "Epoch Gradient Descent" algorithm, presented in Hazan and Kale [2014], that uses the Gradient Descent methodology, but splits the training sample into growing-sized parts (the "epochs"), and makes a restart at the beginning of each epoch:

- changing the parameter;
- using as a starting point the average of the outputs in the previous epoch.

This approach leads to better bounds in the setting of Hazan and Kale [2014] that the "classical conversion" seen previously. In Chapter 4 (Section 4.3), we will apply and adapt this approach to a couple of algorithms, namely Online Newton Step and MetaGrad.

After having seen in this section some ways of transferring algorithms from the online setting to the batch setting, we will focus totally on this latter framework in the next section.

2.4. Regularization in a stochastic batch setting

2.4.1. The framework

In this section, we consider a stochastic batch setting. The goal is to be able to link an output $y \in \mathbb{R}$ to an input x belonging to some probability space χ, by a mapping f included in a set of functions F. Contrary to the individual sequences framework, here we have some knowledge about the way the data is generated. We have access to a sample of N i.i.d. pairs $(x_i, y_i)_{i=1\dots N}$ of random variables, with the same distribution as (x, y). Moreover, one sometimes has access to some additional a priori knowledge about the distribution of the data.

Contrary to the individual sequences setting, more focused on the output y, here the input x has to be taken into consideration.

The performance is still measured by a loss function: $\ell : (w, y) \longmapsto \ell(w, y)$. Looking at the data as random variables, the risk of any mapping f can be defined as:

$$R_f := \mathbb{E}_{x,y}\big[\ell(f(x), y)\big]$$

A first goal is then to construct from the sample $(x_i, y_i)_{i=1\dots N}$ a random mapping $\widehat{f}$ with a risk $R_{\widehat{f}}$ as small as possible: this is called the "prediction problem". We assume that the infimum of $\{R_f, f \in F\}$ is achieved at a function f^* ("the oracle"). The "prediction problem" can then be re-written as trying to compute a mapping $\widehat{f}$ such that the regret (also called excess risk) $R_{\widehat{f}} - R_{f^*}$ is small, either in expectation or with high probability. Even if some tools connect results with high probability to results in expectation (Markov's inequality) and conversely (expectation computed as the integral of the tail of the distribution), the results are not always the same, and some methods can be optimal only for results in expectation, or only for results with high probability. In Chapter 3, we focus on high-probability results.

As the best choice in F in terms of risk is f^*, if the loss is continuous in its first argument, then a mapping $\widehat{f}$ close to f^* may have a small risk (although this depends on the setting and the data). This idea, or sometimes simply the need of further information about f^*, leads to a second problem: the "estimation problem", which consists in computing an estimator $\widehat{f}$ of f^* (as close as possible in expectation or with high probability). Even if these two problems are generally different, in some setups they are equivalent (it is the case for the setup of Chapter 3).

2.4.2. Why a regularization is useful

As our aim is to minimize the expectation of the loss $\mathbb{E}_{x,y}\big[\ell\left(f(x), y\right)\big]$, a natural method is to minimize the empirical average of the loss on the learning sample $\frac{1}{N}\sum_{i=1}^{N} \ell(f(x_i), y_i)$. This approach is called the Empirical Risk Minimization (ERM) method (see, for example, Vapnik [1998] and Koltchinskii [2011]):

$$\widehat{f}_{ERM} \in \operatorname*{argmin}_{f \in F} \frac{1}{N} \sum_{i=1}^{N} \ell\left(f(x_i), y_i\right)$$

However, when the set F is large, then the ERM tends to "overfit", i.e., it sticks too much to the data, and thus it is driven by the data noise, wasting part of its predictive ability because of a lack of generalization. A way to tackle this issue is to add a term $\Psi(f)$, called the "regularization term", to the empirical quantity to minimize. This term is chosen to favour a subset of F, which has desirable properties. Formally, one can thus define the Regularized Empirical Risk Minimization (RERM) method:

$$\widehat{f}_{RERM} \in \operatorname*{argmin}_{f \in F} \frac{1}{N} \sum_{i=1}^{N} \ell\big(f(x_i), y_i\big) + \Psi(f)\cdot \tag{2.4.1}$$

This approach can be interesting for several reasons. First, pushing the prediction towards smaller subsets of F limits the overfitting issue. Second, a well-chosen Ψ, favouring suitable subsets of F in the choice of $\widehat{f}$, may also help. If some characteristics of f^* are known (for instance, if one knows that $\|f^*\|$ is small for some given norm $\|\cdot\|$) then an adequate Ψ allows to focus on subsets of F with such characteristics. It can also be a way of "forcing" a desirable property of the estimator (e.g., sparsity) even if f^* is not assumed to satisfy it.

The influence of the regularization term Ψ can also be seen on the following property (that can not be taken as a definition, due to its recursive aspect): once defined by (2.4.1), $\widehat{f}_{RERM}$ satisfies:

$$\widehat{f}_{RERM} \in \underset{f:\Psi(f)\leqslant\Psi\left(\widehat{f}_{RERM}\right)}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N}\ell\big(f(x_i),y_i\big) \tag{2.4.2}$$

A possible analysis of this expression is that regularization does not change the nature of the minimization to be performed (it is still an empirical risk minimization), but it implicitly changes the set on which this minimization takes place. One of the key points of Chapter 3 will be to modify the LASSO algorithm to ensure that the set $\{f : \Psi(f) \leqslant \Psi(\widehat{f}_{RERM})\}$, unknown in advance, is a relevant set (actually, up to multiplicative constants, the minimal one containing f^*).

2.4.3. Sparsity and ℓ_1 norm

Let us focus on linear regression, in the finite-dimensional vectorial case: one wants to estimate

$$t^* \in \underset{t\in\mathbb{R}^d}{\operatorname{argmin}}\, \mathbb{E}\Big[\ell\left(\langle x,t\rangle, y\right)\Big].$$

A desirable property of an estimator $\widehat{t}$ is sparsity (i.e., the fact of having few non-zero components), at least when it is compatible with a good estimation of t^*. It is particularly useful in high-dimensional setups, because it makes posterior computations easier. This property is also useful in variable selection and interpretation, separating the most impactful variables from those with lesser importance.

A natural way to proceed would be to use what is sometimes called the ℓ_0 "norm": the number of non-zero coefficients (it is not an actual norm, since homogeneity is lacking):

$$\|t\|_0 := \#\big\{i : t_i \neq 0\big\}$$

But $\|.\|_0$ is not a convex function, and the computation of the corresponding RERM is NP-hard (cf. Natarajan [1995]), thus computationally out of reach. So one can use a "convex relaxation" of this problem. Consider the "ℓ_0 ball" composed of vectors with only one non-zero component (and such that this component is bounded by, say, 1); then its convex hull is an ℓ_1 ball. ℓ_1 is the largest ℓ_p norm (and therefore has the smallest unit ball) whose unit ball is convex (cf. Figure 2.1). The idea is then to use a multiple (selected by the statistician) of the ℓ_1 norm for the regularization: $\lambda\|.\|_1$, where λ is to be chosen.

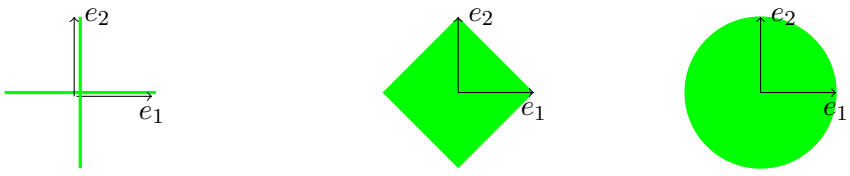


Figure 2.1: Several ℓ_p balls: "$\ell_0 \cap \ell_\infty$ ball", ℓ_1 ball, ℓ_2 ball

In the following, we focus on the case of the square loss: $\ell(x, y) = (x - y)^2$. In this case, the ERM is often called the "least squares estimator", and its ℓ_1 regularized version, already introduced in Section 2.2.3, is called the LASSO ("Least Absolute Shrinkage and Selection Operator"):

$$\widehat{t}_{LASSO} \in \operatorname*{argmin}_{t \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^{N} \big(y_i - \langle x_i, t \rangle\big)^2 + \lambda \|t\|_1$$

As written above, dividing by N the empirical risk is only a question of convention.

Property (2.4.2) becomes:

$$\widehat{t}_{LASSO} \in \operatorname*{argmin}_{t \in \|\widehat{t}_{LASSO}\|_1 B_1^d} \frac{1}{N} \sum_{i=1}^{N} \big(y_i - \langle x_i, t \rangle\big)^2$$

where B_1^d is the unit ball of $\mathbb{R}^d$ for the ℓ_1 norm.

Based on this expression, one can see in Figure 2.2 a geometric heuristic explanation for the sparsity often induced by the LASSO estimator. The ellipses represent the level curves of the empirical risk $t \mapsto \sum_{i=1}^{N} \big(y_i - \langle x_i, t \rangle\big)^2 / N$, the ellipses in green contain at least one element in μB_1^2, and the minimal value within these green level curves is attained at a vertex of μB_1^2, which is therefore a sparse vector. The shape of other unit balls (for instance the ℓ_2 unit ball, which is "round") would make the estimators based on them much more unlikely to be sparse.

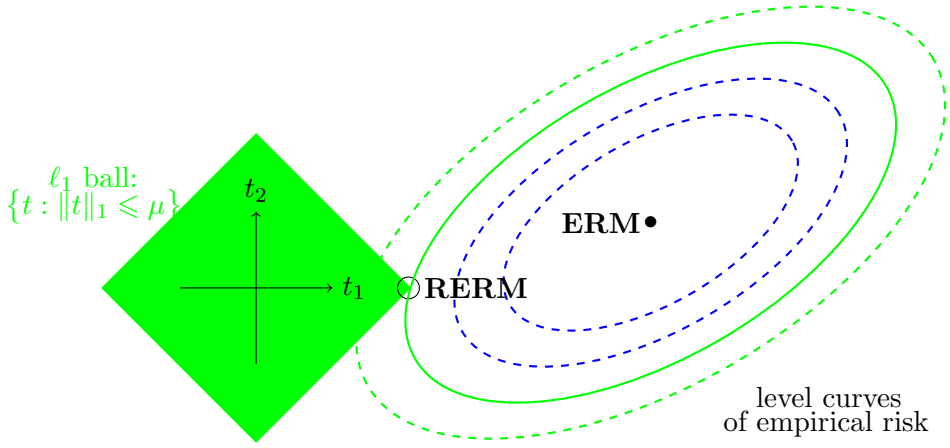


Figure 2.2: Sparsity induced by the ℓ_1 regularization for a two-dimensional estimator

The LASSO has been introduced in Tibshirani [1996] (as a constrained estimator instead of a RERM). Its good performance has made it used in many fields, e.g., genetics (Lu et al. [2011]), ecology (Milns et al. [2010]), electro-chemistry (Saccoccio et al. [2014]), or econometrics (Bai and Ng [2008]).

Other approaches than the ones studied in this thesis rely on the ℓ_1 norm, for instance as a way of selecting the vector with the smallest ℓ_1 norm among all vectors satisfying some

conditions. For example, in the case of an "overcomplete dictionary", with several vectors t satisfying $y_i = \langle x_i, t \rangle$ for all i, the "Basis pursuit" selects the one with the smallest ℓ_1 norm. This method, introduced in Chen et al. [2001], is particularly used in compressed sensing (Foucart and Rauhut [2013]) and in fields related to signal decompositions. In a more classical statistical setup, a method called the "Dantzig selector" (Candes and Tao [2007]) selects some vectors based on the correlation between their residuals and the design matrix, and chooses eventually the selected vector with the smallest ℓ_1 norm.

2.4.4. From model selection to regularization: introducing complexity

We have seen that adding a regularization criterion is a way of favouring the elements of F that behave well with respect to this criterion. If one has defined a set $\{F_m : m \in \mathcal{M}\}$ of models, all included in F, such that $\bigcup_{m \in \mathcal{M}} F_m = F$, one can apply a methodology more obviously "subset-oriented" in its spirit: Model Selection (cf. Massart [2007]). It consists in computing an ERM inside each model F_m:

$$\widehat{f}_m \in \operatorname*{argmin}_{f \in F_m} \frac{1}{N} \sum_{i=1}^{N} \ell\left(f(x_i), y_i\right)$$

and in attributing a penalty $pen\left(F_m\right)$ to each model (this penalty generally increases with the complexity, and therefore often the size, of the model). The final model is selected by balancing the performance of the ERM and the penalty term:

$$\widehat{m} \in \operatorname*{argmin}_{m \in \mathcal{M}} \frac{1}{N} \sum_{i=1}^{N} \ell\left(\widehat{f}_m(x_i), y_i\right) + pen(F_m) \tag{2.4.3}$$

The estimator is then the empirically best element in the chosen subset: $\widehat{f}_{\widehat{m}}$.

This two-step procedure can be re-written as a one-step regularization approach (see Lecué [2011]). Define for each $f \in F$ its minimal penalty: $\text{min\_pen}(f) := \inf_{m : f \in F_m} pen(F_m)$. When this infimum is achieved at some model m, the corresponding model F_m can be seen as the "best-fitted model" for f. Then, the next lemma (Lecué [2011]) shows that min_pen plays the role of a regularization function:

Lemma 2.11.

$$\widehat{f}_{\widehat{m}} \in \operatorname*{argmin}_{f \in F} \frac{1}{N} \sum_{i=1}^{N} \ell\left(f(x_i), y_i\right) + \text{min\_pen}(f)$$

Proof. One can use a *reductio ad absurdum* and assume that there exists f_0 such that:

$$\frac{1}{N} \sum_{i=1}^{N} \ell\left(\widehat{f}_{\widehat{m}}(x_i), y_i\right) + \text{min\_pen}(\widehat{f}_{\widehat{m}}) - \frac{1}{N} \sum_{i=1}^{N} \ell\left(f_0(x_i), y_i\right) + \text{min\_pen}(f_0) > 0.$$

Denote by Δ_0 this gap, and consider a model F_{m_0} such that $f_0 \in F_{m_0}$ and $pen(F_{m_0}) - \text{min\_pen}(f_0) < \Delta_0$. Then by definition of $\widehat{f}_{m_0}$:

$$\frac{1}{N}\sum_{i=1}^{N} \ell\left(\widehat{f}_{m_0}(x_i), y_i\right) \leqslant \frac{1}{N}\sum_{i=1}^{N} \ell\left(f_0(x_i), y_i\right).$$

Moreover, one has $\text{min\_pen}(\widehat{f}_{\widehat{m}}) = pen(F_{\widehat{m}})$ (if not, considering m_1 such that $\widehat{f}_{\widehat{m}} \in F_{m_1}$ and $pen(F_{m_1}) < pen(F_{\widehat{m}})$ would give an immediate contradiction with (2.4.3)).

Therefore:

$$\frac{1}{N}\sum_{i=1}^{N} \ell\left(\widehat{f}_{\widehat{m}}(x_i), y_i\right) + pen(F_{\widehat{m}}) - \frac{1}{N}\sum_{i=1}^{N} \ell\left(\widehat{f}_{m_0}(x_i), y_i\right) + pen(F_{m_0}) > 0.$$

This is a contradiction with the definition of $\widehat{m}$ given in (2.4.3). ■

Conversely, any regularization approach can be seen as a model selection procedure. It suffices to take for models the inverse images of the intervals $]-\infty, r]$, by the regularization function: $\mathcal{M} = \{m_r,\ r \in \mathbb{R}\}$ with $m_r = \Psi^{-1}\big(]-\infty, r]\big)$. In other words, one has to define each model as the set of elements f such that $\Psi(f)$ is inferior to a given value. Then, attributing its canonical penalty $pen(m_r) = r$ to any model m_r and applying the selection model procedure described above, we recover $\widehat{f}_{RERM}$ (see Lemma 3.7.2 in Lecué [2011] for more details).

A difference in the regularization approach and the selection model approach, is that regularization focuses on individual properties of the elements $f \in F$, whereas selection model allows to focus on global properties of entire sets. An important example (as far as overfitting is concerned) is the complexity of the models. There are various ways of measuring complexity of sets, the most suitable ways depending on the context and the data. For finite-dimensional subspaces, the complexity can be their dimension, leading to approaches like Mallows' C_p Mallows [1973], the AIC (Akaike [1974]) or the BIC (Schwarz [1978]). For bounded sets, we may cite the Sudakov complexity, the Talagrand γ-functional, or the Gaussian mean width. The latter, particularly adapted when facing Gaussian or subgaussian data, is defined, for any subset T of $\mathbb{R}^d$ as:

$$\ell^*(T) = \mathbb{E}\Big[\sup_{t\in T}\langle X, t\rangle\Big]$$

where X is a standard Gaussian random variable. Some links and comparisons can be drawn about these complexity measures (for instance, the majorizing measure theorem, or the Sudakov inequality, cf. Ledoux and Talagrand [2013]).

A mid-point between the global complexity of a set F, and individual properties of its elements, is the concept of localized complexity. It is defined as the complexity of the intersection of F and a ball (or a sphere) of a given norm and of arbitrary radius:

$$\mathbb{E}\Big[\sup_{t\in F\cap\rho B}\langle X, t\rangle\Big] \qquad \text{or} \qquad \mathbb{E}\Big[\sup_{t\in F\cap\rho S}\langle X, t\rangle\Big]$$

where B and S denote respectively the unit ball and the unit sphere for the norm of interest. These localized complexities may lead to a sharper analysis on the sets at stake. They will

play a key role in Chapter 3, where we will choose the sets to be intersected with F in a way adapted and "tailored" to the problem at stake, allowing thus an optimal (in a sense that will be defined) regularization.

Chapter 3

Minimax regularization

This chapter is a joint work with Guillaume Lecué.
It has been submitted for publication.

Classical approach to regularization is to design norms enhancing smoothness or sparsity and then to use this norm or some power of this norm as a regularization function. The choice of the regularization function (for instance a power function) in terms of the norm is mostly dictated by computational purpose rather than theoretical considerations.

In this chapter, we design regularization functions that are motivated by theoretical arguments. To that end we introduce a concept of optimal regularization called "minimax regularization" and, as a proof of concept, we show how to construct such a regularization function for the ℓ_1^d norm for the random design setup. We develop a similar construction for the deterministic design setup. It appears that the resulting regularized procedures are different from the one used in the LASSO in both setups.

3.1. Introduction

Let $(\mathcal{X}, \mu)$ be a probability space and (X, Y) be a couple of random variables, in which X is distributed according to μ. One is given a sample of N independent couples $(X_i, Y_i)_{i=1..N}$ distributed according to the joint law of (X, Y). On the basis of this sample, one tries to link X and Y by a random mapping $\widehat{f}$ with $\widehat{f}(X)$ close (in L_2) to Y. This is the classical problem, in learning theory, of the prediction of an output Y from an input X given i.i.d. copies of the couple (X, Y).

To that end, one is given a class F of functions from $\mathcal{X}$ to $\mathbb{R}$ and the aim in learning theory is to mimic the *best* element in F for the prediction of Y by a function of X in F. We assume that F is closed and convex in $L_2(\mu)$ so that it exists a function f^* that minimizes the square loss in F:

$$f^* \in \operatorname*{argmin}_{f \in F} \mathbb{E}\left(Y - f(X)\right)^2. \tag{3.1.1}$$

This function is usually called the oracle (cf. Nemirovski [2000]); it is the closest function in F to Y in L_2. Now, the goal is to construct an estimator $\widehat{f}$ whose $L_2(\mu)$ distance to f^* is as small as possible using the dataset $\{(X_i, Y_i) : i = 1, \ldots, N\}$. In the framework considered in this chapter, the excess risk of $\widehat{f}$, which is the difference $\mathbb{E}(Y - \widehat{f}(X))^2 - \mathbb{E}(Y - f^*(X))^2$, is actually equal to $\|\widehat{f} - f^*\|^2_{L_2(\mu)}$ and so estimating f^* is equivalent to predicting Y; thus we fall back on the original prediction problem by estimating f^* in $L_2(\mu)$.

One may therefore try to bound the quadratic error $\|\widehat{f} - f^*\|_{L_2(\mu)}$ either *in expectation* or *in deviation* with respect to the sample. In this work, we obtain upper bounds on the quadratic error that are valid in deviation, showing that the results are true for "most" samples rather than in average.

Given that we want to be close to a function f^* minimizing $f \to \mathbb{E}(Y - f(X))^2$ over F, a natural candidate for this problem is the Empirical Risk Minimizer (ERM) also known as the "least squares estimator":

$$\widehat{f}_{ERM} \in \operatorname*{argmin}_{f \in F} \frac{1}{N} \sum_{i=1}^{N} (Y_i - f(X_i))^2. \tag{3.1.2}$$

Many works have been carried out for general classes (see, Koltchinskii [2011], Massart [2007], van de Geer [2000], van der Vaart and Wellner [1996]) or on the vectorial case(see Stein [1956] for the famous Stein paradox, and Chatterjee [2014] for elements about the admissibility of the ERM).

It appears that when F is too large (for instance the whole $L_2(\mu)$ space), the ERM tends to "overfit". The understanding of this phenomenon has led to the introduction of "regularization methods" which were originally used to smooth estimators in order to overcome the "overfitting phenomena". Those procedures are nowadays used beyond their smoothing effect and, in particular, they are now extensively used in Statistics and learning theory for their "low-dimensional / sparsity inducing properties". At a high level description, those methods make a trade-of between an "adequation to the data term" and a "regularization term" and

their general form (for the quadratic loss) is

$$\widehat{f} \in \operatorname*{argmin}_{f \in F} \Big(\frac{1}{N} \sum_{i=1}^{N} (Y_i - f(X_i))^2 + \Psi(f) \Big) \tag{3.1.3}$$

where Ψ is a function usually called the *regularization function*.

The "adequation to the data term" can be constructed from any loss function; for the case of the quadratic loss, this term reads like $N^{-1} \sum_{i=1}^{N} (Y_i - f(X_i))^2$.

As for the regularization term Ψ, several choices are possible, enabling to smooth the estimator, or to force a low-dimensional structure. It depends thus on the a priori knowledge one has on the data (and in particular on f^*), and on computational issues.

A first option is the Tikhonov / Ridge regularization:

$$\widehat{f_\lambda} \in \operatorname*{argmin}_{f \in \mathcal{H}} \Big(\frac{1}{N} \sum_{i=1}^{N} (Y_i - f(X_i))^2 + \lambda \|f\|_{\mathcal{H}}^2 \Big) \tag{3.1.4}$$

where $\|\cdot\|_{\mathcal{H}}$ is a Hilbert norm. One expects in this case that $\|f\|_{\mathcal{H}}$ reflects the smoothness of f (for instance $\|f\|_{\mathcal{H}} = f(0) + \big(\int_{\mathbb{R}} |f'(t)|^2 dt \big)^{1/2}$) and that the oracle function f^* has a small $\|f^*\|_{\mathcal{H}}$ norm.

In the finite but high dimensional vectorial case, one often wishes the estimator to be sparse, i.e., to have few non-zero components (for some well-designed basis, cf. Mallat [2009]). This may come from the fact that the vector to be estimated is known in advance to be sparse; or that, in high-dimensional problems, it is computationally important not to have to manage a huge amount of non-zero coefficients.

Then, a natural way to address this question is to use a sparsity-inducing penalization: like the number of non-zero components of the vector, sometimes called the "ℓ_0 norm". Even though it is theoretically appealing (cf. Giraud [2014]), it proves to be computationally intractable (actually NP-Hard, in general, cf. Natarajan [1995]). But for geometric reasons, another regularization is efficient to induce sparsity: the ℓ_1 norm (which can be see as the convex relaxation of the ℓ_0 norm on the unit ℓ_∞-ball). The associated estimator is called the LASSO ("Least Absolute Shrinkage and Selection Operator", Tibshirani [1996]):

$$\widehat{t_\lambda} \in \operatorname*{argmin}_{t \in \mathbb{R}^d} \Big(\frac{1}{N} \sum_{i=1}^{N} (Y_i - \langle X_i, t \rangle)^2 + \lambda \|t\|_1 \Big). \tag{3.1.5}$$

In Candès and Plan [2009], it is emphasized that LASSO leads generally to sparsity, though giving some counter-examples in which it doesn't work well –in particular LASSO struggles when dealing with a data matrix with high correlations among the columns. To tackle this kind of issues, it is possible to "mix" regularizations: it is the principle of the "Elastic net method" (Zou and Hastie [2005]), which penalty is a combination of the ℓ_1 and ℓ_2 norms.

All these methods rely on the choice of one (even two for the Elastic net) regularization parameter λ, fixed by the statistician on the basis of empirical methods such as cross-validation. It has to be chosen wisely in order to make the right trade-off between the two terms and thus to minimize the rate of convergence of the regularization procedure towards the oracle.

3.1.1. Regularization and Model Selection

At this point, it should be clear to the reader that choosing (or even designing) a specific regularization norm like $\|\cdot\|_{\mathcal{H}}$ or $\|\cdot\|_1$ depends much on an a priori knowledge we have. But once this choice has been made, why would someone use the square of this norm in one case (as for the Tikhonov / Ridge regularization), or the norm itself (as for the LASSO) or some other power of this norm (cf. Rohde and Tsybakov [2011] for some examples) in other situations? In many cases, this choice is only made following some computational considerations.

In this work, we want to support the choice of regularization functions (given some norm) on theoretical arguments. To that end we will rely on the model selection theory and, in particular, on a key principle in model selection which is to design penalty functions that capture the "complexity" of a model in the most accurate way.

The right calibration of penalty functions has opened an important stream of researches since the work of Barron et al. [1999]. It has led many researchers to (re)think about the notion of complexity in statistics. In a nutshell, there are mainly three types of quantities that have been introduced to measure the statistical complexity of a statistical model: combinatorial (like the VC dimension, Vapnik [1998]), metric (like the entropy) and random (like Gaussian mean width and Rademacher complexities, Bartlett and Mendelson [2006], Koltchinskii [2006]). Penalty calibration has culminated with the notion of "minimal penalty" that are sharp penalty functions with exact constants (cf. for instance Birgé and Massart [2001]) thanks to second order term analysis of the notion of complexity of a model.

In the present work, we want to put forward the idea that the "right" choice (from a theoretical point of view) of a regularization function may also follow from a careful study of the complexity of a specific family of models. To that end, the first argument is to look at regularization as a model selection problem for which one has to design a sharp penalty. This has been done for instance in Chapter 3.7 in Lecué [2011]. In the particular case where one is given a norm $\|\cdot\|$ for regularization, then the associated regularized ERM is a penalized estimator associated to the sequence of embedded models $(m_r)_{r\geqslant 0}$ where for all $r \geqslant 0$, $m_r = \{f \in F : \|f\| \leqslant r\}$ and the right way to regularize is given by a function $\mathrm{reg} : f \in F \to \mathrm{pen}(m_{\|f\|})$ where $m_{\|f\|}$ is the smallest model in $(m_r)_{r\geqslant 0}$ containing f (see Birgé and Massart [2001]). This idea is a baseline of this work.

Before diving into further details about the way we suggest to construct a regularization function, let us precise what we expect from a good procedure, in particular how we evaluate that a regularization function is the "right" one, at least from a theoretical point of view. We therefore need to introduce a concept of optimality for regularized estimators. Once again we rely on the basics of model selection theory.

Model selection procedures have been used originally to construct adaptive estimators. For the model selection problem we want to solve, this adaptivity problem reads like selecting the smallest model in the family $(\{f \in F : \|f\| \leqslant r\})_{r\geqslant 0}$ containing f^* which is obviously $\{f \in F : \|f\| \leqslant \|f^*\|\}$. Therefore, the adaptation problem we want to solve here is to construct a procedure which performance is as good as if we had been given the value $\|f^*\|$ in advance. In particular, an estimator achieving the minimax rate of convergence over the model $\{f \in F : \|f\| \leqslant \|f^*\|\}$ would solve this adaptation problem. In what follows, we design regularization functions in order to meet this requirement, but before that, we clarify the

notion of minimax rate over a model for the type of deviation results we prove below.

To simplify the exposition, we will focus on a specific, though very classical and widely-used, framework: the vectorial case, i.e. when $F = \left\{\langle \cdot, t\rangle : t \in T\right\}$ is a class of linear functionals from $\mathbb{R}^d$ to $\mathbb{R}$ indexed by some subset $T \subset \mathbb{R}^d$, with Gaussian design, and Gaussian noise (with known variance σ^2).

Definition 3.1.1. *Let $T \subset \mathbb{R}^d$, X denote a standard Gaussian vector in $\mathbb{R}^d$ and ξ be a centered real-valued Gaussian random variable with variance σ^2, independent of X. For all $t^* \in T$, define the random variable $Y^{t^*} = \langle X, t^*\rangle + \xi$ and denote by $\mathcal{Y}^T := \{Y^{t^*} : t^* \in T\}$ the set of all such random variables.*

*Let $\widehat{t}_N$ be a statistics from $(\mathbb{R}^d \times \mathbb{R})^N$ to $\mathbb{R}^d$. Let $0 < \delta_N < 1$ and $\zeta_N > 0$. We say that $\widehat{t}_N$ **performs with accuracy** ζ_N **and confidence** $1 - \delta_N$ **relative to the set of targets** $\mathcal{Y}^T$, if for all $Y^{t^*} \in \mathcal{Y}^T$, with probability, w.r.t. to a sample $\mathcal{D} := \{(X_i, Y_i) : i = 1, \cdots, N\}$ of i.i.d. copies of (X, Y), at least $1 - \delta_N$, $\|\widehat{t}_N - t^*\|_2^2 \leqslant \zeta_N$.*

*We say that $\mathcal{R}_N$ **is a minimax rate of convergence over** T **for the confidence** $1 - \delta_N$ if the two following statements hold:*

1. *there exists a statistics $\widehat{t}_N$ which performs with accuracy $\mathcal{R}_N$ and confidence $1 - \delta_N$ relative to the set of targets $\mathcal{Y}^T$*
2. *there exists an absolute constant $g_0 > 0$ such that if $\tilde{t}_N$ is a statistics which attains an accuracy ζ_N with confidence $1 - \delta_N$ relative to the set of targets $\mathcal{Y}^T$ then necessarily $\zeta_N \geqslant g_0 \mathcal{R}_N$.*

In the following (cf. Theorem 3.1.3 below), we recall a result from Lecué and Mendelson [2013] on the minimax rate of convergence over $T = \rho B_1^d$, the unit ℓ_1^d-ball of radius $\rho \geqslant 0$, for a constant confidence (i.e., for instance, $\delta_N = 1/4$). Note that classical minimax rates of convergence are usually given in expectation (cf. for instance Tsybakov [2009]). The main difference here with Definition 3.1.1 is that it is given for deviation results: the minimax rate $\mathcal{R}_N$ may depend on the confidence parameter δ_N (cf. Lecué and Mendelson [2013]).

In the present work, we are interested in procedures achieving the minimax rate of convergence over the model $\{t \in \mathbb{R}^d : \|t\| \leqslant \|t^*\|\}$. This provides a natural way to introduce a notion of optimality for the problem of designing regularization functions.

Definition 3.1.2. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^d$, $0 < \delta_N < 1$ and $T \subset \mathbb{R}^d$. Let us consider the following RERM for some function $\Psi : \mathbb{R}_+ \to \mathbb{R}$:*

$$\widehat{t} \in \operatorname*{argmin}_{t \in \mathbb{R}^d} \left(\frac{1}{N} \sum_{i=1}^N (Y_i - \langle X_i, t\rangle)^2 + \Psi(\|t\|) \right)$$

*constructed from a sample $\mathcal{D} := \{(X_i, Y_i) : i = 1, \cdots, N\}$ of i.i.d. copies of (X, Y^{t^*}) where $Y^{t^*} = \langle X, t^*\rangle + \xi$ with $X \sim \mathcal{N}(0, I_{d\times d})$, $\xi \sim \mathcal{N}(0, \sigma^2)$ is independent of X and $t^* \in \mathbb{R}^d$. We say that Ψ is a **minimax regularization function for the norm** $\|\cdot\|$ **and the confidence** $1 - \delta_N$ **over** T, if there exists an absolute constant $g_1 > 0$ such that for all $t^* \in T$, the RERM $\widehat{t}$ is such that with $\mathbb{P}_{t^*}$-probability at least $1 - \delta_N$, $\|\widehat{t} - t^*\|_2^2 \leqslant g_1 \mathcal{R}_N^{\|t^*\|_1}$, where $\mathcal{R}_N^{\|t^*\|_1}$ is the minimax rate of convergence over $\{t \in \mathbb{R}^d : \|t\| \leqslant \|t^*\|\}$ and $\mathbb{P}_{t^*}$ denotes the probability distribution of a N sample of i.i.d. copies of (X, Y^{t^*}).*

The aim of this work is to show that one can design minimax regularization functions by finding the right notion of complexity of the sequence of embedded models $\big(\{t \in \mathbb{R}^d : \|t\| \leqslant r\}\big)_{r \geqslant 0}$. Note however that there should be some situations where designing such an optimal regularization function would be impossible at some given confidence parameter δ_N. In particular, such a situation should happen when the Empirical risk minimization (ERM) procedure over the "true model" $\{t \in \mathbb{R}^d : \|t\| \leqslant \|t^*\|\}$ is not itself a minimax procedure over the model $\{t \in \mathbb{R}^d : \|t\| \leqslant \|t^*\|\}$. This happens for constant confidence bound (for instance, when $\delta_N = 1/4$) when there is a gap in Sudakov inequality (cf. Lecué and Mendelson [2013] for more details). Nevertheless, in Lecué and Mendelson [2013], it is proved that for high confidence bounds (that is when δ_N decays exponentially fast with the complexity of the model) ERM is always minimax over convex classes. It appears that for the case of ℓ_1^d-balls ERM is minimax for all confidence regime therefore this subtlety will not show up in this special case.

3.1.2. General approach provided in this chapter

Let us now present our approach. As we mentioned before, we want to construct a regularization function depending on the complexity of the models $\{t \in \mathbb{R}^d : \|t\| \leqslant r\}$ for all $r \geqslant 0$. This leads to choose a RERM (in the vectorial case) having the following form:

$$\widehat{t} \in \operatorname*{argmin}_{t \in \mathbb{R}^d} \Big(\frac{1}{N} \sum_{i=1}^{N} (Y_i - \langle X_i, t \rangle)^2 + \operatorname{comp}\big(\|t\| B_{\|\cdot\|}\big) \Big) \tag{3.1.6}$$

where $B_{\|\cdot\|}$ is the unit ball associated with the given regularization norm $\|\cdot\|$ and for all $t \in \mathbb{R}^d$, $\|t\| B_{\|\cdot\|} = \{u \in \mathbb{R}^d : \|u\| \leqslant \|t\|\}$. The key feature in (3.1.6) is the "complexity function" $r \geqslant 0 \to \operatorname{comp}\big(r B_{\|\cdot\|}\big)$ which aims at measuring with the best possible accuracy the complexity of the models $r B_{\|\cdot\|}$ for all $r \geqslant 0$ from a statistical point if view. Or course, finding the right notion of complexity is paramount in this approach.

To aim at optimality, we advocate complexities that are tailored for the specific statistical problem at stake. It turns out that the "right" choice of complexity, and hence, of regularization, is linked to the behavior of two empirical processes. Those two empirical processes are ultimately connected to the two sources of statistical complexities in the considered problem. When estimating t^* from the data $(X_i, Y_i)_{i=1}^N$ there are two statistical issues:

- 1)(an inverse problem) t^* is observed only through $\mathbb{X}$, where $\mathbb{X} \in \mathbb{R}^{N \times d}$ is the operator whose rows vectors are given by the X_i's;
- 2)(noisy data) the observations have been corrupted by some noise ξ.

The action of the operator $\mathbb{X}$ on the models $r B_{\|\cdot\|}$ for all $r \geqslant 0$ plays a prominent role in our analysis. In particular, the size of the intersection of its kernel with the model is a natural minimax lower bound for any estimator since any two vectors in the kernel of $\mathbb{X}$ and the model are indistinguishable. The effect of the "distortion" of the operator $\mathbb{X}$ does not show up for small models (i.e. small values of r) because of the presence of the noise which blurs everything at small scales. But passing beyond some threshold for the signal-to-noise ratio r/σ, only the

distortion of $\mathbb{X}$ matters from a statistical point of view. This phenomenon occurs only when $N \lesssim d$, because that is the regime where $\mathbb{X}$ has a non trivial kernel. On the contrary, in the low dimensional setup $d \lesssim N$, $\mathbb{X}$ is well conditioned with high probability and therefore, there is no statistical complexity coming from the distortion of $\mathbb{X}$ since in that regime there is no such distortion. Controlling the distortion of $\mathbb{X}$ is a key issue in high-dimensional statistics. It is behind all classical properties like RIP (cf. Candès and Tao [2010]) or REC (cf. Bickel et al. [2009]) and it will play equally a key role in our analysis. In particular, the ℓ_2^d diameter of the intersection of $\mathbb{X}$ with the model $rB_{\|\cdot\|}$ will appear explicitly in the optimal regularization. Given that $\mathbb{X}$ is a standard Gaussian random matrix, this diameter will be the Gelfand width of $rB_{\|\cdot\|}$ in our example (cf. Pinkus [1985], Chapter 2 in Chafaï et al. [2012] or Lecué and Mendelson [2013] for more details on Gelfand widths and their role in signal processing and learning theory).

3.1.3. Overview of the chapter; main results

As a proof of concept we present an example of the construction of a minimax regularization function in the popular set-up of regularization by the ℓ_1^d norm. Let us recall the statistical model we used in both Definition 3.1.2 and Definition 3.1.1: the Gaussian linear regression model with a Gaussian design

$$Y = \langle X, t^* \rangle + \xi \tag{3.1.7}$$

where $X \sim \mathcal{N}(0, I_{d\times d})$ and $\xi \sim \mathcal{N}(0, \sigma^2)$ are independent, centered Gaussian variables in $\mathbb{R}^d$ and $\mathbb{R}$ respectively. As written before, a dataset $(X_i, Y_i)_{i=1}^N$ of N i.i.d. copies of the couple (X, Y) is provided and one wants to use it to estimate t^*.

Note that we choose a Gaussian random design to make the exposition as simple as possible. The results can be extended to more general sub-Gaussian designs. Nonetheless, our goal is not to provide general results but to show that the approach we present allows to achieve minimax regularization in some classical set-up. Moreover, we want to see the effect of the random design on the construction of a minimax regularization.

In the supplementary material 3.4, we consider a fixed design setup, in which one still has for all i: $Y_i = \langle X_i, t^* \rangle + \xi_i$ with $\xi_i \sim \mathcal{N}(0, \sigma^2)$ i.i.d., but with X_i deterministic, satisfying an "isomorphic property" on "compressible vectors", equivalent to the RIP from Candès and Tao [2010]. We will see that under this property, the arguments and the results will be quite similar to the random design case.

We will not be interested in getting optimal or sharp numerical constants, and some of the inequalities and coefficients in the arguments will be rather loose from this point of view, but what actually matters is that the quantity will have the right order of magnitude w.r.t. N, d, σ and $\|t^*\|_1$.

As for our choice to consider regularizations that are functions of the ℓ_1^d norm, its motivation is that the ℓ_1^d-norm has been one of the most studied regularization norm since the beginning of high-dimensional statistics, in particular for the reasons presented in this Introduction. Moreover, as mentioned previously, the ERM over ℓ_1^d-balls is minimax for every confidence $1 - \delta_N$ (cf. Lecué and Mendelson [2013]), this makes the construction of minimax regularization possible for different deviation parameters and this makes the exposition also simpler.

Let the choice of the ℓ_1^d-norm as a regularization norm be made once and for all. Now, the problem we want to solve is to construct a regularization function $\Psi : \mathbb{R} \to \mathbb{R}$ such that the regularized procedure

$$\widehat{t}_\Psi \in \operatorname*{argmin}_{t\in\mathbb{R}^d} \Big(\frac{1}{N}\sum_{i=1}^N (Y_i - \langle X_i, t\rangle)^2 + \Psi(\|t\|_1) \Big) \tag{3.1.8}$$

achieves the minimax rate of convergence over $\|t^*\|_1 B_1^d$ given N i.i.d. data $(X_i, Y_i), i = 1, \dots, N$ distributed according to (3.1.7). And, we want $\widehat{t}_\Psi$ to satisfy that property whatever $t^* \in \mathbb{R}^d$ is.

Now, let us explain the strategy we use to design a minimax regularization function. We denote for all $\rho \geqslant 0$ and $r \geqslant 0$, $\rho B_1^d = \{t \in \mathbb{R}^d : \|t\|_1 \leqslant \rho\}$, and $rB_2^d = \{t \in \mathbb{R}^d : \|t\|_2 \leqslant r\}$. The starting point to our approach is that $\widehat{t}_\Psi$ minimizes $t \mapsto P_N \mathcal{L}_t^\Psi$ over $\mathbb{R}^d$, where, for every $t \in \mathbb{R}^d$,

$$P_N \mathcal{L}_t^\Psi := \left(\frac{1}{N}\sum_{i=1}^N (Y_i - \langle X_i, t\rangle)^2 + \Psi(\|t\|_1) \right) - \left(\frac{1}{N}\sum_{i=1}^N (Y_i - \langle X_i, t^*\rangle)^2 + \Psi(\|t^*\|_1) \right),$$

in particular, $P_N \mathcal{L}_{\widehat{t}}^\Psi \leqslant P_N \mathcal{L}_{t^*}^\Psi = 0$. So if one shows that $P_N \mathcal{L}_t^\Psi > 0$ for all $\|t\|_1 \gtrsim \|t^*\|_1$ then this will prove that $\|\widehat{t}_\Psi\|_1 \lesssim \|t^*\|_1$, proving that $\widehat{t}_\Psi$ belongs to the right model. This will be essentially the main step since for the correct choice of Ψ, we will show that the regularization has no effect within the right model and that the RERM $\widehat{t}^\Psi$ has essentially the same statistical behavior as the ERM in $\|t^*\|_1 B_1^d$ which is known to be minimax. We will therefore conclude that $\widehat{t}^\Psi$ can learn t^* at the minimax rate of convergence within the model $\|t^*\|_1 B_1^d$ without knowing in advance the radius $\|t^*\|_1$.

Using the quadratic / multiplier decomposition as in Lecué and Mendelson [2013], Saumard [2012, 2017], one can write $P_N \mathcal{L}_t^\Psi$ as the sum of three terms: $P_N \mathcal{L}_t^\Psi = P_N \mathcal{Q}_{t-t^*} + P_N \mathcal{M}_{t-t^*} + \mathcal{R}_{t,t^*}$ where

- $P_N \mathcal{Q}_{t-t^*} := \sum_{i=1}^N \left(\langle X_i, t^*\rangle - \langle X_i, t\rangle \right)^2 / N$ is the "quadratic process"
- $P_N \mathcal{M}_{t-t^*} := 2\sum_{i=1}^N \left(Y_i - \langle X_i, t^*\rangle \right) \left(\langle X_i, t^*\rangle - \langle X_i, t\rangle \right) / N$ is the "multiplier process"
- $\mathcal{R}_{t,t^*} := \Psi(\|t\|_1) - \Psi(\|t^*\|_1)$ is the regularization part.

The definition of our complexity will thus be a consequence of the study of the behavior of the quadratic and multiplier empirical processes indexed by $t \in \rho B_1^d$ for all $\rho \geqslant 0$. The two processes are associated to the two statistical complexities previously discussed:

- 1) the quadratic process can be written as $P_N \mathcal{Q}_{t-t^*} = \|\mathbb{X}(t - t^*)\|_2^2$ and is well behaved (i.e. of the order of $\|t - t^*\|_2^2$) when $\mathbb{X}$ is well conditioned;
- 2) the multiplier process is measuring the statistical complexity coming from the noise $\xi = Y - \langle X, t^*\rangle$, $P_N \mathcal{M}_{t-t^*}$ is the empirical correlation between the noise and the model shifted by $\langle \cdot, t^*\rangle$.

All the game is now to identify regions of the space $\mathbb{R}^d$ where the statistical complexity comes from the distortion of $\mathbb{X}$ or from the noise. This drives the construction of the optimal regularization function Ψ.

In order to identify those regions, note that for every fixed $t \in \mathbb{R}^d$, the distribution of these two processes depend on $t - t^*$ only by its ℓ_2^d-norm $\|t - t^*\|_2$, in two different ways: $P_N \mathcal{Q}_{t-t^*}$ in a quadratic way, $P_N \mathcal{M}_{t-t^*}$ in a linear way. So it is natural to partition the model ρB_1^d into vectors with "small" ℓ_2 norm – i.e. the intersection of $\rho B_1^d \cap r B_2^d$ for an adequate radius r – and vectors of ρB_1^d with ℓ_2^d-norm larger than r. We will see that outside rB_2^d, with high probability the two processes are "well-behaved" and regularization is unnecessary; but inside rB_2^d it is not the case, the operator $\mathbb{X}$ may have a kernel and the noise is making the estimation hard: hence, this is where the regularization will be needed to keep control of the situation and this is precisely where the regularization function is designed. In that case, either the statistical complexity comes from the size of the intersection of the kernel of $\mathbb{X}$ with ρB_1^d and therefore one needs to take $\Psi(\rho)$ of the order of this diameter (which appears to be equal to the Gelfand width of ρB_1^d to the square) or the statistical complexity comes from the noise and then $\Psi(\rho)$ is of the order of the oscillations of the multiplier process inside $\rho B_1^d \cap r B_2^d$.

The choice of the "adequate radius" r is of course paramount in our approach. It results from the right understanding of the two previously discussed sources of statistical complexities: the bigger these complexities, the bigger this radius (since, as we mentioned, outside rB_2^d the processes are well-behaved). First, we want to identify the smallest ℓ_2^d radius $r_Q(\rho)$ above which $\mathbb{X}$ is well-behaved in ρB_1^d, i.e. such that for every $t \in \rho B_1^d$, if $\|t - t^*\|_2 \geqslant r_Q(\rho)$, then $P_N \mathcal{Q}_{t-t^*} = \|\mathbb{X}(t - t^*)\|_2^2 \sim \|t - t^*\|_2^2$. Then, we need to identify the smallest ℓ_2^d-radius $r_M(\rho)$ above which the effect of the noise is below the signal intensity that is above which one can clearly identify if $t \neq t^*$ when $\|t - t^*\|_2 \geqslant r_M(\rho)$. To that end we want to make the oscillations of the multiplier process smaller than the one of the quadratic process, which is of the order of $\|t - t^*\|_2^2$ when $\|t - t^*\|_2 \geqslant r_Q(\rho)$. It will appear that, in our framework, the two radii obtained from the above trade-offs are solution of fixed point equations for all $\rho \geqslant 0$: for some absolute constants Q and η (to be chosen later):

- the "quadratic fixed point" is $r_Q(\rho) := \inf\left(r > 0 : \ell^*(\rho B_1^d \cap r B_2^d) = Qr\sqrt{N}\right)$
- the "multiplier fixed point" is $r_M(\rho) := \inf\left(r > 0 : \sigma \ell^*(\rho B_1^d \cap r B_2^d) = \eta r^2 \sqrt{N}\right)$

where $\ell^*(\rho B_1^d \cap r B_2^d)$ is the Gaussian mean width of the localized set $\rho B_1^d \cap r B_2^d$ defined as

$$\ell^*(\rho B_1^d \cap r B_2^d) = \mathbb{E} \sup_{t \in \rho B_1^d \cap r B_2^d} \langle G, t \rangle$$

where G is a standard Gaussian vector in $\mathbb{R}^d$. As our framework involves "Gaussian randomness" in both the design and the noise, it is not surprising that the Gaussian mean width arise when dealing with the control of the two processes. However, Gaussian mean widths appear in learning theory, statistics and signal processing way beyond the "full Gaussian framework" as considered here (see, for instance, Lecué and Mendelson [2017]).

These two fixed points have been introduced in Lecué and Mendelson [2013] and used later in Lecué and Mendelson [2016] for the study or ERM and RERM. As their names suggest, the quadratic fixed point will be used to control the quadratic process, and the multiplier fixed point to control the multiplier process. Their general definitions use an inequality rather than an equality inside the infimum: $r_Q(\rho)$ is defined as $\inf\left(r>0 : \ell^*(\rho B_1^d \cap r B_2^d) \leqslant Qr\sqrt{N}\right)$ and $r_M(\rho)$ as $\inf\left(r>0 : \sigma\ell^*(\rho B_1^d \cap r B_2^d) \leqslant \eta r^2\sqrt{N}\right)$. This allows to deal with infinite-dimensional set-ups in which the mapping $r \mapsto \ell^*(\mathcal{F}\cap rB_2)$ is not necessarily continuous. But in our case, this mapping is continuous and the infimum is attained in a point for which there is exact equality.

It appears that one can provide an explicit formulation for the two fixed point $r_Q(\rho)$ and $r_M(\rho)$ in many situations and, in particular, in the case of the ℓ_1^d-norm (cf. Lecué and Mendelson [2013]): for some absolute constants $C_M^{(1)}, C_M^{(2)}, C_Q^{(1)}, C_Q^{(2)}$ and $\zeta < 1 < \zeta'$, for all ρ, there exists $C_M \in [C_M^{(1)}, C_M^{(2)}]$ such that:

$$r_M^2(\rho) = C_M \begin{cases} \frac{\sigma^2 d}{N} & \text{if } \rho^2 N \geqslant \sigma^2 d^2 \\ \rho\sigma\sqrt{\frac{1}{N}\log\left(\frac{e\sigma d}{\rho\sqrt{N}}\right)} & \text{if } \sigma^2 \log d \leqslant \rho^2 N \leqslant \sigma^2 d^2 \\ \rho\sigma\sqrt{\frac{\log(ed)}{N}} & \text{if } \rho^2 N \leqslant \sigma^2 \log d. \end{cases} \tag{3.1.9}$$

and there exists $C_Q \in [C_Q^{(1)}, C_Q^{(2)}]$ such that

$$r_Q^2(\rho) = C_Q \begin{cases} 0 & \text{if } N \geqslant \zeta' d \\ \frac{\rho^2}{N}\log\left(\frac{ed}{N}\right) & \text{if } N \leqslant \zeta d. \end{cases} \tag{3.1.10}$$

Note that when $\zeta d \leqslant N \leqslant \zeta' d$, $r_Q(\rho)$ decays from $(\rho^2/N)\log(ed/N)$ to 0 and one only has an upper estimate on $r_Q(\rho)$ given by $C_Q\rho^2/N$. We will therefore not consider this case in the following since it involves to deal with sharp estimates on the spectra of squared or approximatively squared Gaussian random matrices. Note also that C_M and C_Q may depend on ρ, N, d, σ but they are both controlled from above and below by absolute constants (independent of ρ, N, d and σ).

Now that we have a way to measure the statistical complexity of a model we need one more thing before turning to the effective construction of a minimax regularization for the ℓ_1^d-norm: we need to know the minimax rate of convergence over ℓ_1^d-ball ρB_1^d for all $\rho \geqslant 0$. We will see below that one way to measure the statistical complexity of a model is closely related to its minimax rate. To that end, we summarize the main results in the constant deviation case $\delta_N = 1/4$ from section 4.1 in Lecué and Mendelson [2013] in the following theorem.

Theorem 3.1.3. *Consider the Gaussian linear model with Gaussian design introduced in (3.1.7). Let $\rho > 0$. The minimax rate of convergence for constant confidence parameter $\delta_N = 1/4$ over ρB_1^d is achieved by the ERM and is given (up to absolute constants) by*

$$\min\left(r^2(\rho), \rho^2\right) \text{ where } r(\rho) = \max\left(r_Q(\rho), r_M(\rho)\right). \tag{3.1.11}$$

Up to multiplicative absolute constants, this rate is given for some $\zeta < 1 < \zeta'$,

1. *when $N \leqslant \log d$, by ρ^2,*
2. *when $\log d \leqslant N \leqslant \zeta d$, by*

$$\begin{cases} \rho^2 & \text{if } \rho^2 N \leqslant \sigma^2 \log d, \\ \rho\sigma\sqrt{\frac{1}{N}\log\left(\frac{ed^2\sigma^2}{\rho^2 N}\right)} & \text{if } \sigma^2 \log d \leqslant \rho^2 N \leqslant \frac{\sigma^2 N^2}{\log(ed/N)}, \\ \frac{\rho^2}{N}\log\left(\frac{ed}{N}\right) & \text{if } \rho^2 N \geqslant \frac{\sigma^2 N^2}{\log(ed/N)} \end{cases}$$

3. *when $N \geqslant \zeta' d$, by*

$$\begin{cases} \rho^2 & \text{if } \rho^2 N \leqslant \sigma^2 \log d, \\ \rho\sigma\sqrt{\frac{1}{N}\log\left(\frac{ed^2\sigma^2}{\rho^2 N}\right)} & \text{if } \sigma^2 \log d \leqslant \rho^2 N \leqslant \sigma^2 d^2, \\ \frac{\sigma^2 d}{N} & \text{if } \rho^2 N \geqslant \sigma^2 d^2. \end{cases}$$

In other words, for all $\rho \geqslant 0$ and $t^* \in \rho B_1^d$, the ERM $\widehat{t}_\rho^{ERM} \in \operatorname{argmin}_{t\in\rho B_1^d} \sum_{i=1}^N (Y_i - \langle X_i, t\rangle)^2$, is such that, with probability at least $3/4$, $\|\widehat{t}_\rho^{ERM} - t^*\|_2^2 \leqslant \min\left(r^2(\rho), \rho^2\right)$. Moreover, there are no estimator that can do uniformly better than the ERM $\widehat{t}_\rho^{ERM}$ over ρB_1^d when $N \notin (\zeta d, \zeta' d)$.

Note that we have decided to present the result in the constant deviation result (that is for $\delta_N = 1/4$) whereas it is actually true with a much better probability estimate in section 4.1 in Lecué and Mendelson [2013]. We will also obtain our main results with an exponentially large deviation below.

As mentioned previously, when $N \in [\zeta d, \zeta' d]$, we only have an upper bound on $(r_Q(\rho))^2$ that does not match the minimax lower bound. As a consequence, the $N \sim d$ regime is not considered in Theorem 3.1.3. Notable is that the rate ρ^2 is the trivial rate obtained by taking the ℓ_2^d diameter of the model ρB_1^d which is simply 2ρ. Therefore, any statistics $\tilde{t}_N$ (like the ERM $\widehat{t}_\rho^{ERM}$) taking its values in ρB_1^d satisfies with probability 1, $\|\tilde{t}_N - t^*\|_2^2 \leqslant 4\rho^2$ for all $t^* \in \rho B_1^d$. This is a trivial bound that one can get for free as long as the radius ρ is known. However, for the construction of an optimal regularization function which can be seen as an adaptation to the radius $\|t^*\|_1$, which is therefore not known, this trivial bound is not available. This will be an issue for designing a minimax regularization function when $\|t^*\|_1$ is unknown and small (actually smaller than $\sigma\sqrt{\log(ed)/N}$). Somehow the "signal-to-noise ratio" is too small for the models ρB_1^d with small ρ's. Therefore, the trivial upper bound ρ^2 is optimal when ρ is known but in the other case we will have to pay the price due to the noise and there will be no way to achieve the trivial optimal ρ^2 bound for small ρ's (except for the trivial estimator $\widehat{t}_0 = 0$, see the discussion after Proposition 3.1.6). That is the reason why we will not be able to construct a minimax regularization function over the entire space $\mathbb{R}^d$ but only for t^* such that $\|t^*\|_1 \gtrsim \sigma/\sqrt{\log(ed)/N}$. We will also show that such a construction of an optimal regularization function over the entire space $\mathbb{R}^d$ is actually not possible at all later in Proposition 3.1.6.

Finally let us turn to the construction of a minimax regularization function for the ℓ_1^d-norm. To that end we will use the function $\rho \geqslant 0 \mapsto r^2(\rho) = \max\left(r_Q^2(\rho), r_M^2(\rho)\right)$ as a sharp way to measure the complexity of the model ρB_1^d. The main result of this article is that this function is a minimax regularization function as introduced in Definition 3.1.2.

Theorem 3.1.4. *There are absolute constants $\eta, Q, \zeta, \zeta', \Delta_0, c_0$ such that the following holds. When $\zeta' d \geqslant N$ or $\zeta d \leqslant N$, a minimax regularization function for the ℓ_1^d-norm over $\mathbb{R}^d \backslash (\Delta_0 \sigma \sqrt{\log(ed)/N} B_1^d)$ for the confidence parameter $\delta_N = 1/4$ is given by the following function: for all $\rho > 0$,*

$$\Psi(\rho) = c_0 r^2(\rho)$$

where $r(\rho) = \max\left(r_Q(\rho), r_M(\rho)\right)$ for

$$r_Q(\rho) = \inf\left(r > 0 : \ell^*(\rho B_1^d \cap r B_2^d) = Q r \sqrt{N}\right)$$

and

$$r_M(\rho) = \inf\left(r > 0 : \sigma \ell^*(\rho B_1^d \cap r B_2^d) = \eta r^2 \sqrt{N}\right)$$

denoting by $\ell^*(\rho B_1^d \cap r B_2^d)$ the Gaussian mean width of the localized sets $\rho B_1^d \cap r B_2^d$. In that case, the rate achieved by the RERM $\widehat{t}_\Psi$ is the minimax rate $r^2(\|t^*\|_1)$ when $\|t^*\|_1 \geqslant \Delta_0 \sigma \sqrt{\log(ed)/N}$.

The shape of the minimax regularization function $\rho \to \Psi(\rho) = c_0 r^2(\rho)$ is given in Figure 3.1 in the two cases $N \leqslant \zeta d$ ("high-dimensional statistics") and $N \geqslant \zeta' d$ ("classical or low-dimensional statistics").

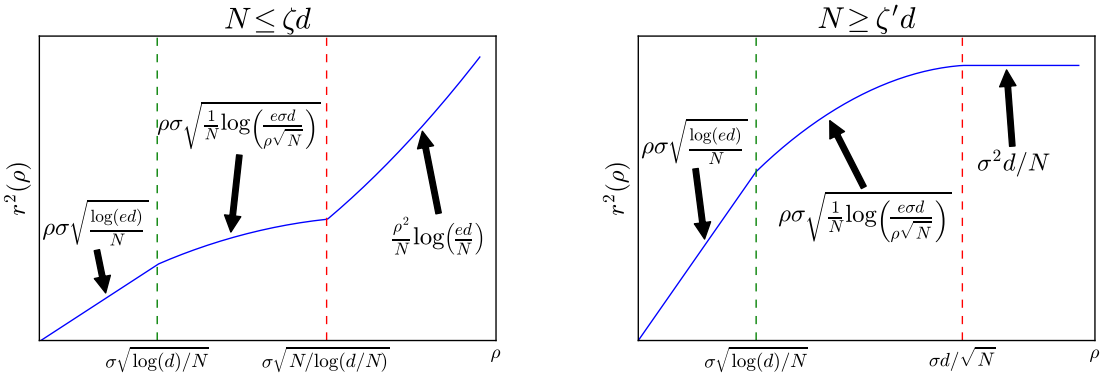


Figure 3.1: Shape of the graph of the minimax regularization function $\rho \to r^2(\rho)$ of the ℓ_1^d-norm for the cases $N \leqslant \zeta d$ (left) and $N \geqslant \zeta' d$ (right)

The only difference between the two cases ($\zeta' d \geqslant N$ or $\zeta d \leqslant N$) appears for large radii ρ. The reason for that lies in the statistical complexity coming (or not) from the distortion of the operator $\mathbb{X}$. In the low-dimensional case, $\mathbb{X}$ is such that (with high probability), $\|\mathbb{X}t\|_2 \sim \|t\|_2$ for all $t \in \mathbb{R}^d$. There is no distortion coming from $\mathbb{X}$. Somehow observing $\mathbb{X}t^*$ is the same as

observing t^* itself, one just has to invert $\mathbb{X}$ – this can be done because $\mathbb{X}$ acts like an isomorphy on the entire space $\mathbb{R}^d$. Therefore, there is no statistical complexity coming from $\mathbb{X}$ and so its associated complexity parameter $r_Q(\cdot)$ does not show up in the final complexity parameter $r(\cdot) = \max(r_M(\cdot), r_Q(\cdot))$. We therefore end up with $r(\cdot) = r_M(\cdot)$ in the low-dimensional case. In particular, for large radii ρ (for which, one has $\rho B_1^d \cap r_M(\rho) B_2^d = r_M(\rho) B_2^d$), we pay the worst rate of convergence in $\mathbb{R}^d$, which is $\sigma^2 d/N$ because learning over ρB_1^d for large values of ρ is as hard as learning over the entire space $\mathbb{R}^d$ and the price for the latter is the rate $\sigma^2 d/N$.

The situation is totally different in the high-dimensional setup because in that case the operator $\mathbb{X} \in \mathbb{R}^{N \times d}$ has a non-trivial kernel; therefore, observing $\mathbb{X}t^*$ is totally different from observing t^* (for instance, imaging that $t^* \in \ker \mathbb{X}$). This adds to the statistical complexity of the problem of estimating t^*. In this regime, both the noise and the distortion effect of $\mathbb{X}$ appear in the statistical complexity of the estimation problem; this means that both complexity parameter $r_Q(\cdot)$ and $r_M(\cdot)$ appear in the total complexity parameter $r(\cdot)$ and therefore in the ultimately designed minimax regularization function. For small values of ρ, the effect of the noise is predominant but for large values of ρ this is the effect of $\mathbb{X}$ which is the main responsible of the statistical complexity. In particular, the ℓ_2^d-diameter of $\ker \mathbb{X} \cap \|t^*\|_1 B_1^d$ is important because there is of course no way to distinguish t^* from $t^* + h$ for all $h \in \mathbb{R}^d$ such that $\mathbb{X}t^* = \mathbb{X}(t^* + h)$ that is for all $h \in \ker \mathbb{X}$ such that $t^* + h \in \|t^*\|_1 B_1^d$. Hence, estimating t^* is at least as hard as estimating any point in $(t^* + \ker \mathbb{X}) \cap \|t^*\| B_1^d$ and therefore, no estimator $\tilde{t}$ can estimate t^* at a rate better than $\operatorname{diam}\left(\ker \mathbb{X} \cap \|t^*\|_1 B_1^d, \ell_2^d\right)^2$. The latter quantity is itself lower bounded by the Gelfand's N-width of $\|t^*\| B_1^d$ defined as

$$c_N(\|t^*\|_1 B_1^d) := \inf\left\{\operatorname{diam}(\ker \Gamma \cap \|t^*\|_1 B_1^d) : \Gamma \in \mathbb{R}^{N \times d}\right\} \sim \|t^*\| \min\left\{1, \sqrt{\frac{\log(ed/N)}{N}}\right\} \tag{3.1.12}$$

the latter result is due to Garanaev and Gluskin Garnaev and Gluskin [1984]. It appears that the Gelfand's N-width of $\|t^*\| B_1^d$ are achieved (up to absolute constants and with high probability) by the kernel of standard $N \times d$ Gaussian matrices, which is exactly the case of the design matrix $\mathbb{X}$. Therefore, with high probability,

$$\operatorname{diam}\left(\ker \mathbb{X} \cap \|t^*\|_1 B_1^d, \ell_2^d\right)^2 \sim c_N^2(\|t^*\|_1 B_1^d) \sim \frac{\|t^*\|_1^2}{N} \log\left(\frac{ed}{N}\right) \sim r_Q^2(\|t^*\|_1) \tag{3.1.13}$$

This is exactly the price we pay in $r_Q(\rho)$ when $\rho \geqslant \sigma\sqrt{N/\log(ed/N)}$. That is the reason why we take the regularization function $\Psi(\rho)$ of the order of the Gelfand's N-width of ρB_1^d (to the square) for large radii ρ: it is the right concept of statistical complexity that shows up in this part of the space $\mathbb{R}^d$, where the statistical complexity coming from the distortion of $\mathbb{X}$ becomes more important than the one due to the noise.

Remark 3.1.5 (Regularization function for the LASSO)**.** *The LASSO is the RERM procedure obtained for a linear regularization function $\Psi(\rho) = \sigma\rho\sqrt{\log d/N}$ which is obtained by using a trivial upper bound on the complexity of the model ρB_1^d in (3.1.8):*

$$\operatorname{comp}\left(\rho B_1^d\right) = r_M^2(\rho) \leqslant \frac{\sigma \ell^*\left(\rho B_1^d\right)}{\sqrt{N}} = \sigma\rho\sqrt{\frac{\log(ed)}{N}}. \tag{3.1.14}$$

This complexity is obtained by simply removing the localization (i.e. the intersection with rB_2^d) in the multiplier process when computing $r_M(\cdot)$, and does not take $r_Q(\cdot)$ into account. This means that the distortion of the operator $\mathbb{X}$ is assumed to have no effect on the statistical complexity of the problem. This is why estimation results for the LASSO deal only with the reconstruction of vectors which are sparse or almost sparse, i.e. for vectors belonging to the cone appearing in the RE or CC conditions, cf. Bickel et al. [2009]. Over this cone, the quadratic process behaves nicely (that is, the isomorphic property from Proposition 3.2.1 holds on this cone) or in other words, the operator $\mathbb{X}$ is well-conditioned on the set of vectors we want to reconstruct, so that there is no statistical complexity coming from the distortion of this operator. So, as long as estimation of sparse or approximately sparse vectors is concerned, there is no need for the complexity function $r_Q(\cdot)$. That is why the regularization function used for the LASSO take into account only the fixed point $r_M(\cdot)$ associated to the statistical complexity due to the noise and not the one from the inverse problem. On the contrary, by taking $r_M(\cdot)$ and $r_Q(\cdot)$ into account, our regularization function allows us to deal with the full space $\mathbb{R}^d$ (except for a small ℓ_1^d-ball centered in 0, cf. Proposition 3.1.6) and not only a cone.

Moreover, as said before, the way the regularization function is designed in (3.1.14) is sub-optimal because it uses a trivial upper bound on $r_M(\cdot)$ instead of using the exact formulation of $r_M(\cdot)$ as in (3.1.9). Contrary to the LASSO, this latter exact formulation takes into account, thanks to the localization, the fact that the regularization is not needed on the whole space –in some areas the random processes behave nicely whatever. The suboptimal approach for the LASSO is likely to be responsible for a loss in the rate of convergence achieved by the LASSO, which is $s\sigma^2\log(ed)/N$ whereas the minimax rate is $s\sigma^2\log(ed/s)/N$ (cf. Bellec et al. [2016]). This is not a big loss, especially when $d >> s$, but from a purely theoretical point of view the right way to regularize for the reconstruction of sparse vectors should be using $r_M^2(\|t\|_1)$ instead of $\sigma\|t\|_1\sqrt{\log(ed)/N}$ as it is the case for the LASSO. However, the resulting regularization function would be concave (cf. the right-hand side plot in Figure 3.1). Therefore, the small price paid from a theoretical point of view by using the trivial upper bound in (3.1.14) seems to be worth the computational gain obtained by using a convex regularization as does the LASSO.

Let us now turn to the adaptation problem in the ball ρB_1^d for $\rho \sim \sigma\sqrt{\log(ed)/N}$. We want to answer the following question: is it possible to construct a regularization function $\Psi(\cdot)$ so that the associated regularized procedure $\widehat{t}_\Psi$ is adaptive on the entire space $\mathbb{R}^d$? Or (even stronger) is there any statistic that can be adaptive (in the sense that it achieves the rate of the ERM on $\|t^*\|_1 B_1^d$ without knowing $\|t^*\|_1$ beforehand) on the entire space $\mathbb{R}^d$ (this statistics may not be a regularized procedure)? It appears that the answer to this question is negative, which we prove in the following proposition.

However, one needs to be cautious with the next statement because there is a trivial estimator $\widehat{t}_0 = 0$ such that for every $t^* \in \mathbb{R}^d$, with probability 1, $\|\widehat{t}_0 - t^*\|_2^2 = \|t^*\|_2^2$ and therefore $\widehat{t}_0$ is adaptive on ρB_1^d as long as the minimax rate over ρB_1^d is ρ^2, which is the case for any $\rho \lesssim \sigma\sqrt{\log(ed)/N}$. Therefore, there exists a procedure adaptive on ρB_1^d when $\rho \sim \sigma\sqrt{\log(ed)/N}$. Moreover, according to Theorem 3.1.4, there exists a procedure adaptive on $\mathbb{R}^d \backslash \rho B_1^d$. But the question concerns the adaptation on the *entire space* $\mathbb{R}^d$ *at the same time.*

The following statement shows that if $\widehat{t}$ is a procedure adaptive on $\mathbb{R}^d \backslash \rho B_1^d$ then it cannot

be adaptive on ρB_1^d for $\rho \sim \sigma\sqrt{\log(ed)/N}$. Moreover, it also proves that adaptation on the entire space $\mathbb{R}^d$ is not possible and that Theorem 3.1.4 is optimal given that the range of radii $[\Delta_0\sigma\sqrt{\log(ed)/N}, +\infty)$ on which it is adaptive cannot be inflated (up to absolute constants). Before turning to the statement let us denote by $\mathbb{P}_{t^*}$ the probability distribution of a N-sample $(X_i, Y_i)_{i=1}^N$ of i.i.d. copies of (X, Y) when (X, Y) is distributed according to (3.1.7).

Proposition 3.1.6. *Assume that $2d \geqslant \exp(544/225)$ and that there exists an absolute constant χ_1 such that the following holds. Let $\rho \leqslant 2\sigma\sqrt{(\log(2d))/(96N)}$ be such that $16\chi_1 r^2(\rho) \leqslant \rho^2$ and denote by $(e_j)_{j=1}^d$ the canonical basis of $\mathbb{R}^d$. Assume that $\widehat{t}$ is an estimator such that for every $t^* \in \{\pm\rho e_1, \dots, \pm\rho e_d\}$,*

$$\mathbb{P}_{t^*}\left[\|\widehat{t} - t^*\|_2^2 \leqslant \chi_1 r^2(\rho)\right] \geqslant \frac{3}{4}.$$

Then, for every $t^* \in (\rho/2)B_1^d$,

$$\mathbb{P}_{t^*}\left[\|\widehat{t} - t^*\|_2^2 \geqslant \rho^2/16\right] \geqslant \frac{1}{2}. \tag{3.1.15}$$

The proof of Proposition 3.1.6 is given in Section 3.3. Note that the only property of the design X used to prove Proposition 3.1.6 is isotropicity. Since isotropicity does not tell much on the distortion properties of the design matrix $\mathbb{X}$, it means that Proposition 3.1.6 is only based on the statistical complexity coming from the noise. This is not a surprise given that Proposition 3.1.6 is a result for very small radii less than $\sim \sigma\sqrt{\log(ed)/N}$. At that scale, even if $\ker\mathbb{X}$ is in the worst possible position, i.e. $\operatorname{diam}(\ker\mathbb{X} \cap \rho B_1^2, \ell_2^d)^2 = \operatorname{diam}(\rho B_1^2, \ell_2^d)^2 = \rho^2$, we still have $\rho^2 \lesssim r_M^2(\rho)$. Hence, the distortion of $\mathbb{X}$ does not play any role at this very small scale and therefore that is not a surprise that Proposition 3.1.6 is true for any isotropic design X.

Finally, let us rephrase Proposition 3.1.6 in other words. Proposition 3.1.6 shows that if a procedure can learn all vectors in $\{\pm\rho e_1, \dots, \pm\rho e_d\}$ at the minimax rate $r^2(\rho)$ then this estimator cannot learn any $t^* \in (\rho/2)B_1^d$ at the optimal minimax rate ρ^2 for confidence $1/4$. For instance, given that the result (3.1.15) holds for any $t^* \in (\rho/2)B_1^d$, in particular, for $t^* = 0$, it tells that $\widehat{t}$ cannot estimate $t^* = 0$ at a rate better than $\rho^2 \sim \sigma^2\log(ed)/N$ whereas the minimax rate over $\rho^* B_1^d$ for $\rho^* = 0$ is obviously 0. Finally, note that the condition $16\chi_1 r^2(\rho) \leqslant \rho^2$ implies that $\rho \gtrsim \sigma\sqrt{\log(ed)/N}$ so that the phase transition radius above which adaptation is possible but not below is of the order of $\sigma\sqrt{\log(ed)/N}$ which is the radius we have found in Theorem 3.1.4.

3.2. Proof of Theorem 3.1.4

Most of the proof consists in showing that with high probability, $\widehat{t}$ belongs to $t^* + \rho^* B_1^d$ where

$$\rho^* = \max\left(10, 8\frac{(C_M^{(2)})^2}{(C_M^{(1)})^2\eta} + 1\right)\|t^*\|_1. \tag{3.2.1}$$

(where the value of η will be fixed later). Once this goal is achieved, it is straightforward to show (again with high probability) that $\|\widehat{t}-t^*\|_2^2$ is less than the minimax rate of convergence over $\rho^* B_1^d$.

To do so, we will prove that with high probability, any t outside $t^* + \rho^* B_1^d$ satisfies $P_N \mathcal{L}_t^\Psi > 0$ (whereas $P_N \mathcal{L}_{\widehat{t}}^\Psi \leqslant 0$). We partition $\mathbb{R}^d \backslash (t^* + \rho^* B_1^d)$ into shelves of the form $t^* + (2^{j+1}\rho^* B_1^d \backslash (2^j \rho^*) B_1^d)$, in which the regularization function remains mostly constant. We only need to study the smallest shells, i.e. for $k = 1, \dots, K_0$ for some well-chosen K_0 (K_0 is the smallest integer so that $2^{K_0-1}\rho^* B_1^d \cap r(2^{K_0-1}\rho^*) B_2^d = r(2^{K_0-1}\rho^*) B_2^d$), the part of $\mathbb{R}^d$ for which $\|t\|_1 \geqslant 2^{K_0}\rho^*$ will be treated by an homogeneity argument.

On each of the smallest shelves, the argument is roughly and heuristically the following: we place ourselves on a high probability event on which random processes ($P_N \mathcal{M}, P_N \mathcal{Q}$, their supremum, their infimum,...) "behave nicely" (i.e. they both scale in a favourable way with respect to $\|t - t^*\|_2^2$). Then, as $P_N \mathcal{M}_{t-t^*}$ is the only possibly negative term, it suffices to identify zones where $P_N \mathcal{Q}_{t-t^*} > P_N \mathcal{M}_{t-t^*}$ (then directly $P_N \mathcal{L}_t^\Psi > 0$), and compensate $|P_N \mathcal{M}_{t-t^*}|$ on the other part by using a penalty that is close to the supremum (on this other part) of $P_N \mathcal{M}_{t-t^*}$. As $P_N \mathcal{Q}_{t-t^*}$ grows quicker than $P_N \mathcal{M}_{t-t^*}$ with respect to $\|t - t^*\|_2$, the big zone where $P_N \mathcal{Q}_{t-t^*} > P_N \mathcal{M}_{t-t^*}$ will be the exterior of $t^* + rB_2^d$ for an adequate r (cf. Figure 3.2). This r must be such that any t in the exterior of this ball satisfies $P_N \mathcal{Q}_{t-t^*} \gtrsim \|t - t^*\|_2^2 \gtrsim P_N \mathcal{M}_{t-t^*}$, and we will see that the first inequality amounts to $r \geqslant r_Q(\rho)$, and the second one to $r \geqslant r_M(\rho)$. Next, the supremum of $P_N \mathcal{M}_{t-t^*}$ on $\rho B_1^d \cap r(\rho) B_2^d$ is less than $r_M(\rho) r(\rho) \leqslant r^2(\rho)$ for $r(\rho) = \max(r_M(\rho), r_Q(\rho))$. We therefore set the regularization function $\Psi(\rho)$ at level ρ to be proportional to the quantity $r^2(\rho)$ because it is this quantity measuring the amplitude of the oscillation of the multiplier process in $\rho B_1^d \cap r(\rho) B_2^d$.

As for its presentation, the proof of Theorem 3.1.4 is divided into two parts. The first part (Section 3.2.1) defines the event on which the two processes "behave nicely" and computes a lower bound on its probability. In the second part (Section 3.2.2) we will place ourselves on this event and carry out the deterministic geometric part of the argument.

3.2.1. Probabilistic control of the processes

Instead of controlling the two processes on shelves, we will control them on the full ℓ_1 balls, because it does not change the complexity, up to constants, and the very last step of the proof requires a control on the two processes on the full ℓ_1 ball $\rho^* B_1^d$.

Control of the quadratic process

This first section provides the classical analysis of the quadratic process based upon its isomorphic properties on the set of "almost sparse vectors". Such a property holds in the optimal regime of observation (or the optimal size of the cone of "almost sparse vectors"), only in the sub-Gaussian case. It is the case we are considering here since we assumed that the design is a standard Gaussian random variable. This analysis borrows some ideas from the "isomorphic method" from Bartlett and Mendelson [2006] or the Restricted Isometry Property from Candès et al. [2006] in the sub-Gaussian case. For the sake of completeness we recall here the argument from Lecué and Mendelson [2013].

Proposition 3.2.1. *There are absolute constants C_1 and C_1' such that the following holds. Let $X_1, \dots, X_N$ be N i.i.d. standard Gaussian vectors in $\mathbb{R}^d$. Denote by Ω^* the event on which: for every $\rho \geqslant \rho^*$ and all $t \in t^* + \rho B_1^d$,*

$$\text{if } \|t - t^*\|_2 \geqslant r_Q(\rho) \text{ then } \frac{1}{2}\|t - t^*\|_2^2 \leqslant \frac{1}{N}\sum_{i=1}^N \langle X_i, t - t^*\rangle^2 \leqslant \frac{3}{2}\|t - t^*\|_2^2. \tag{3.2.2}$$

Then, one has $\mathbb{P}[\Omega^*] \geqslant 1 - 2\exp(-C_1 Q^2 N)$ as long as $Q \leqslant C_1'$.

Proof. First note that for all $\rho > 0$, $r_Q(\rho) = \rho r_Q(1)$. Indeed, we have

$$\ell^*(\rho B_1^d \cap r B_2^d) = \ell^*\left(\rho(B_1^d \cap (r/\rho)B_2^d)\right) = \rho \ell^*\left(B_1^d \cap (r/\rho)B_2^d\right)$$

and so

$$\begin{aligned} r_Q(\rho) &= \inf\{r > 0 : \ell^*(\rho B_1^d \cap r B_2^d) = Q\sqrt{N} r\} \\ &= \inf\{r > 0 : \ell^*(B_1^d \cap (r/\rho) B_2^d) = Q\sqrt{N}(r/\rho)\} \\ &= \rho \ \inf\{r > 0 : l^*(B_1^d \cap r B_2^d) = Q\sqrt{N} r\} = \rho r_Q(1). \end{aligned} \tag{3.2.3}$$

For all $\rho > 0$, define the event $\Omega(\rho)$ on which one has for all $t \in t^* + \rho B_1^d$,

$$\text{if } \|t - t^*\|_2 \geqslant r_Q(\rho) \text{ then } \frac{1}{2}\|t - t^*\|_2^2 \leqslant \frac{1}{N}\sum_{i=1}^N \langle X_i, t - t^*\rangle^2 \leqslant \frac{3}{2}\|t - t^*\|_2^2.$$

Let us show that if $\Omega(\rho^*)$ holds then for any $\rho \geqslant \rho^*$, $\Omega(\rho)$ holds as well. Assume that $\Omega(\rho^*)$ holds. Consider $t \in t^* + \rho B_1^d$ such that $\|t - t^*\|_2 > r_Q(\rho)$ and define

$$t' := t^* + (\rho^*/\rho)(t - t^*) \in t^* + \rho^* B_1^d.$$

It follows from (3.2.3) that $r_Q(\rho^*) = (\rho^*/\rho) r_Q(\rho)$. Thus $\|t' - t^*\|_2 = (\rho^*/\rho)\|t - t^*\|_2 > (\rho^*/\rho) r_Q(\rho) = r_Q(\rho^*)$, and since $\Omega(\rho^*)$ holds, it follows that $\|t' - t^*\|_2^2/2 \leqslant \sum_{i=1}^N \langle X_i, t' - t^*\rangle^2/N \leqslant 3\|t' - t^*\|_2^2/2$. This implies that $\|t - t^*\|_2^2/2 \leqslant \sum_{i=1}^N \langle X_i, t - t^*\rangle^2/N \leqslant 3\|t - t^*\|_2^2/2$ so $\Omega(\rho)$ holds.

As a conclusion, $\Omega^* = \Omega(\rho^*)$ and we can now lower bound the probability that this event holds.

Let us consider the class of linear functions

$$F = \left\{\langle \cdot, t - t^*\rangle, t \in t^* + \rho^* B_1^d \cap r_Q(\rho^*) S_2^{d-1}\right\} = \left\{\langle \cdot, t\rangle, t \in \rho^* B_1^d \cap r_Q(\rho^*) S_2^{d-1}\right\}.$$

We assume that F is non empty (if $F = \emptyset$ then the theorem is trivially satisfied). It follows from Theorem 1.12 in Mendelson [2016] that for any $x > 0$, with probability at least $1 - 2\exp\big(-C_1 \min(x^2, x\sqrt{N})\big)$,

$$\sup_{f \in F}\left|\frac{1}{N}\sum_{i=1}^N f^2(X_i) - \mathbb{E} f^2(X)\right| \leqslant C_2\left(\frac{\Delta\gamma}{\sqrt{N}} + \frac{\gamma^2}{N} + \frac{x\Delta^2}{\sqrt{N}}\right)$$

where Δ is the diameter in ψ_2 of F and γ is Talagrand's γ_2 functional of F w.r.t. ψ_2. Note that since X is a standard Gaussian variable in $\mathbb{R}^d$, for any $t \in \mathbb{R}^d$, $\|\langle X, t\rangle\|_{\psi_2} = C_3\|t\|_2$ for some absolute constant C_3. It follows that

$$\Delta = 2 \sup_{t \in \rho^* B_1^d \cap r_Q(\rho^*) S_2^{d-1}} C_3\|t\|_2 = 2C_3 r_Q(\rho^*) \text{ and } \gamma = \gamma_2(\rho^* B_1^d \cap r_Q(\rho^*) S_2^{d-1}, \ell_2^d).$$

Moreover, it follows from the Majorizing measure theorem (cf. Chapter 1 in Talagrand [2014]) that

$$\gamma_2(\rho^* B_1^d \cap r_Q(\rho^*) S_2^{d-1}, \ell_2^d) \leqslant C_4 \ell^*(\rho^* B_1^d \cap r_Q(\rho^*) S_2^{d-1})$$

Since F is non-empty, by Lemma 3.3.1 the right-hand side is equal to $C_4\ell^*(\rho^* B_1^d \cap r_Q(\rho^*) B_2^{d-1})$ and so by definition of $r_Q(\rho^*)$, one has $\gamma \leqslant C_4 Q r_Q(\rho^*)\sqrt{N}$.

Since X is isotropic (i.e. for any $t \in \mathbb{R}^d$, $\mathbb{E}\langle X, t\rangle^2 = \|t\|_2^2$), we obtain for $x = Q\sqrt{N}$ that, with probability greater than $1 - 2\exp(-C_1 \min(Q, Q^2)N)$, for any $t \in t^* + \rho^* B_1^d \cap r_Q(\rho^*) S_2^{d-1}$,

$$\left| \frac{1}{N} \sum_{i=1}^N \langle X_i, t - t^*\rangle^2 - \|t - t^*\|_2^2 \right| \leqslant (2C_2C_3C_4Q + C_2C_4^2Q^2 + 4C_2QC_3^2) r_Q^2(\rho^*)$$

So, as long as long as: $Q \leqslant C_1' := \min\{1, (12C_2C_3C_4)^{-1}, (\sqrt{6C_2}C_4)^{-1}, (24C_2C_3^2)^{-1}\}$, one has, with probability greater than $1 - 2\exp(-C_1Q^2N)$, for all $t \in t^* + \rho^* B_1^d \cap r_Q(\rho^*) S_2^{d-1}$,

$$\left| \frac{1}{N} \sum_{i=1}^N \langle X_i, t - t^*\rangle^2 - \|t - t^*\|_2^2 \right| \leqslant \frac{r_Q^2(\rho^*)}{2} = \frac{1}{2}\|t - t^*\|_2^2. \tag{3.2.4}$$

In other words, the quadratic process satisfies an isomorphic property on the set $t^* + (\rho^* B_1^d \cap r_Q(\rho^*) S_2^{d-1})$. Now, it remains to extend this result to the set of vectors $t \in t^* + \rho^* B_1^d$ such that $\|t - t^*\|_2 \geqslant r_Q(\rho^*)$. Let t be such a vector and define $t' := t^* + (r_Q(\rho^*)/\|t - t^*\|)(t - t^*)$. Since $t' \in t^* + (\rho^* B_1^d \cap r_Q(\rho^*) S_2^{d-1})$, it satisfies the isomorphic property from (3.2.4) and so

$$\frac{1}{2}\|t - t^*\|_2^2 \leqslant \frac{1}{N} \sum_{i=1}^N \langle X_i, t - t^*\rangle^2 \leqslant \frac{3}{2}\|t - t^*\|_2^2$$

which corresponds exactly to the event $\Omega(\rho^*)$. Therefore, $\mathbb{P}[\Omega^*] = \mathbb{P}[\Omega(\rho^*)] \geqslant 1 - 2\exp(-C_1Q^2N)$. ■

Control of the multiplier process

In this section, we provide a control of the multiplier process on several shelves of the space $\mathbb{R}^d$.

Define r_0 as the non-null solution to $\sigma\ell^*(rB_2^d) = \eta r^2\sqrt{N}$ – i.e. $r_0 = \sigma\ell^*(B_2^d)/(\eta\sqrt{N}) = (\sigma/\eta)\sqrt{d/N}$. Let ρ_0 be the smallest ρ such that ρB_1^d contains $r_0 B_2^d$ –i.e., $\rho_0 = r_0\sqrt{d}$. We can see that ρ_0 is such that $r_M(\rho) = r_M(\rho_0)$ for all $\rho \geqslant \rho_0$. Indeed, one first sees that $r_M(\rho_0) = r_0$, since $\sigma\ell^*(r_0 B_2^d \cap \rho_0 B_1^d) = \sigma\ell^*(r_0 B_2^d) = \eta r_0^2\sqrt{N}$, and $\sigma\ell^*(rB_2^d \cap \rho_0 B_1^d) = \sigma r\ell^*(B_2^d) > \eta r^2\sqrt{N}$

for all $r < r_0$ and for $r > r_0$, $\sigma\ell^*(rB_2^d \cap \rho_0 B_1^d) \leqslant \sigma r \ell^*(B_2^d) \leqslant \eta r^2\sqrt{N}$. This last argument also holds for $\rho \geqslant \rho_0$. In the latter case, if $r > r_0$ then

$$\sigma\ell^*(rB_2^d \cap \rho B_1^d) \leqslant \frac{r}{r_0}\sigma\ell^*(r_0 B_2^d \cap \rho B_1^d) = \frac{r}{r_0}\sigma\ell^*(r_0 B_2^d \cap \rho_0 B_1^d) = \frac{r}{r_0}\eta r_0^2\sqrt{N} < \eta r^2\sqrt{N}$$

which means that $r_M(\rho) \leqslant r_0$. And as $\rho \geqslant \rho_0$, $r_M(\rho) \geqslant r_M(\rho_0) = r_0$. Therefore, for $\rho \geqslant \rho_0$, $r_M(\rho)$ is constant, equal to r_0. And on $[0, \rho_0]$, r_M is non-decreasing: let $\rho' \leqslant \rho'' \leqslant \rho_0$, then $\sigma\ell^*(r_M(\rho')B_2^d \cap \rho'' B_1^d) \geqslant \sigma\ell^*(r_M(\rho')B_2^d \cap \rho' B_1^d) = \eta r_M(\rho')^2\sqrt{N}$ so $r_M(\rho'') \geqslant r_M(\rho')$.

We denote $K_0 = \min\{k \in \mathbb{N} : 2^k\rho^* \geqslant 2\rho_0\}$: we will see later that K_0 is defined that way to be the number of the first "shell" such that $r_M(2^{K_0-1}\rho^*)B_2^d \subset 2^{K_0-1}\rho^* B_1^d$.

Proposition 3.2.2. *There exists an absolute constant C_5 such that the following holds. Let $X_1, \dots, X_N$ be N i.i.d. standard Gaussian vectors in $\mathbb{R}^d$ and $\xi_1, \dots, \xi_N$ be N standard real-valued Gaussian variables independent of the X_i's. For all $k = 0, \dots, K_0$, denote by A_k the event on which, for every $t \in \mathbb{R}^d$ such that $\|t - t^*\|_1 \leqslant 2^k\rho^*$:*

$$|P_N\mathcal{M}_{t-t^*}| \leqslant \frac{1}{4}\max\left(r_M(2^k\rho^*)^2, \|t - t^*\|_2^2\right). \tag{3.2.5}$$

Then, for $\eta = 1/(16\sqrt{2})$, one has

$$\mathbb{P}\left[\bigcap_{k=0}^{K_0} A_k\right] \geqslant 1 - 2\exp\left(-C_5 N\right) - 40\exp\left(-\frac{C_M^{(1)} N r_M(\rho^*)^2)}{1024 C_M^{(2)}\sigma^2}\right)$$

when $\rho^* \geqslant 4096\log(2)\sigma/\left(C_M^{(1)}\sqrt{N}\right)$.

Proof. We first work conditionally to the $\xi_i, i = 1, \dots, N$. Let $\rho > 0$ and define $T(\rho) := t^* + \rho B_1^d \cap r_M(\rho)B_2^d$. It follows from the Gaussian concentration inequality (cf. Borell's inequality in Ledoux [2001]) that, for all $x > 0$, with probability greater than $1 - 2\exp(-x^2/2)$,

$$\left|\sup_{t\in T(\rho)}\sum_{i=1}^N \xi_i\langle X_i, t - t^*\rangle - \mathbb{E}\sup_{t\in T(\rho)}\sum_{i=1}^N \xi_i\langle X_i, t - t^*\rangle\right| \leqslant x\sigma(T(\rho))$$

where

$$\sigma(T(\rho)) = \sup_{t\in T(\rho)}\sqrt{\mathbb{E}\left(\sum_{i=1}^N \xi_i\langle X_i, t - t^*\rangle\right)^2}.$$

Conditionally to the ξ_i's, the Gaussian process $\left(\sum_{i=1}^N \xi_i\langle X_i, t - t^*\rangle\right)_{t\in T(\rho)}$ has the same distribution as the Gaussian process $\left(\widehat{\sigma_N}\langle X_1, t - t^*\rangle\right)_{t\in T(\rho)}$ for $\widehat{\sigma_N} := \sqrt{\sum_{i=1}^N \xi_i^2}$. This yields

$$\mathbb{E}\left[\sup_{t\in T(\rho)}\sum_{i=1}^N \xi_i\langle X_i, t - t^*\rangle\right] = \widehat{\sigma_N}\ell^*(T(\rho) - t^*) = \widehat{\sigma_N}\ell^*(\rho B_1^d \cap r_M(\rho)B_2^d) = \widehat{\sigma_N}\frac{\eta\sqrt{N}r_M^2(\rho)}{\sigma}$$

and $\sigma(T(\rho)) = \sup_{t\in T(\rho)} \widehat{\sigma_N}\sqrt{\mathbb{E}[\langle X, t-t^*\rangle^2]} = \widehat{\sigma_N}\sup_{t\in T(\rho)}\|t-t^*\|_2 \leqslant \widehat{\sigma_N} r_M(\rho)$.

So, conditionally on $(\xi_i)_{i=1}^N$, for all $x>0$, one has, with probability at least $1-2\exp(-x^2/2)$,

$$\sup_{t\in T(\rho)}\left|\sum_{i=1}^N \xi_i\langle X_i, t-t^*\rangle\right| \leqslant \widehat{\sigma_N}\frac{\eta\sqrt{N}r_M^2(\rho)}{\sigma} + x\widehat{\sigma_N} r_M(\rho)$$

Thus, taking $x = \eta\sqrt{N}r_M(\rho)/\sigma$ in the previous statement, one gets, on an event which probability is at least $1-2\exp(-\eta^2 N r_M^2(\rho)/(2\sigma^2))$,

$$\sup_{t\in t^*+\rho B_1^d \cap r_M(\rho)B_2^d}\left|\frac{1}{N}\sum_{i=1}^N \xi_i\langle X_i, t-t^*\rangle\right| \leqslant 2\eta\frac{\widehat{\sigma_N}}{\sigma\sqrt{N}} r_M^2(\rho). \tag{3.2.6}$$

It remains to prove, on the same event, the result for all $t\in t^*+\rho B_1^d$ such that $\|t-t^*\|_2 > r_M(\rho)$. Define $t' := t^* + \big(r_M(\rho)/\|t-t^*\|_2\big)(t-t^*)$. Since $t'\in T(\rho)$, it follows from (3.2.6) that (on the same event):

$$\left|\frac{1}{N}\sum_{i=1}^N \xi_i\langle X_i, \frac{r_M(\rho)}{\|t-t^*\|_2}(t-t^*)\rangle\right| = \left|\frac{1}{N}\sum_{i=1}^N \xi_i\langle X_i, t'-t^*\rangle\right| \leqslant 2\eta\frac{\widehat{\sigma_N}}{\sigma\sqrt{N}} r_M(\rho)^2$$

and since $r_M(\rho)\leqslant\|t-t^*\|_2$ one gets

$$\left|\frac{1}{N}\sum_{i=1}^N \xi_i\langle X_i, t-t^*\rangle\right| \leqslant 2\eta\frac{\widehat{\sigma_N}}{\sigma\sqrt{N}}\|t-t^*\| r_M(\rho) \leqslant 2\eta\frac{\widehat{\sigma_N}}{\sigma\sqrt{N}}\|t-t^*\|^2.$$

Hence, with probability (conditionally to the ξ_i) at least $1-2\exp\big(-\eta^2 N r_M(\rho)^2)/(2\sigma^2)\big)$, the multiplier process is controlled such that

$$\sup_{t\in t^*+\rho B_1^d}|P_N\mathcal{M}_{t-t^*}| \leqslant 4\eta\frac{\widehat{\sigma_N}}{\sigma\sqrt{N}}\max\left(r_M(\rho)^2, \|t-t^*\|_2^2\right). \tag{3.2.7}$$

A control of the probability measure of the event A_k follows by applying the previous result to $\rho = 2^k\rho^*$ when $\eta\leqslant 1/(16\sqrt{2})$ together with a control of the term $\widehat{\sigma}_N$. It follows from an union bound that, conditionally to the ξ_i, (3.2.7) is satisfied for all $\rho = 2^k\rho^*, k=0,\cdots,K_0$ on an event whose probability measure is larger than

$$1-2\sum_{k=0}^{K_0}\exp\left(-\eta^2 N r_M(2^k\rho^*)^2)/(2\sigma^2)\right). \tag{3.2.8}$$

We handle the last term below thanks to Lemma 3.3.2.

Now, we handle the random variables $\xi_1,\ldots,\xi_N$. It appears that only a control of the empirical variance term $\widehat{\sigma}_N/\sqrt{N}$ is needed to get a fully deterministic upper bound in the right-hand term of (3.2.7). It follows from Bernstein inequality for subexponential variables (cf. Theorem 1.2.7 in Chafaï et al. [2012]) that with probability greater than $1-2\exp(-C_5 N)$,

$$\left|\frac{1}{N}\sum_{i=1}^N \xi_i^2 - \sigma^2\right| \leqslant \sigma^2,$$

which implies $\widehat{\sigma_N}/\sqrt{N} \leqslant \sqrt{2}\sigma$. Therefore, for $\eta = 1/(16\sqrt{2})$ we have

$$\mathbb{P}\left[4\eta\widehat{\sigma_N}/(\sqrt{N}\sigma) \leqslant 1/4\right] \geqslant 1 - 2\exp\left(-C_5 N\right). \tag{3.2.9}$$

Binding together (3.2.7), (3.2.8) and (3.2.9) gives

$$\mathbb{P}\left[\bigcap_{k=0}^{K_0} A_k\right] \geqslant 1 - 2\exp\left(-C_5 N\right) - 2\sum_{k=0}^{K_0} \exp\left(-\frac{N r_M(2^k\rho^*)^2}{1024\sigma^2})\right).$$

Finally, Lemma 3.3.2 yields the following bound

$$\sum_{k=0}^{K_0} \exp\left(-\frac{N r_M(2^k\rho^*)^2)}{1024\sigma^2}\right) \leqslant \frac{10}{1 - \exp\left(-\frac{C_M^{(1)}\sqrt{N}\rho^*}{4096\sigma}\right)} \exp\left(-\frac{C_M^{(1)}}{C_M^{(2)}}\frac{N r_M\left(\rho^*\right)^2}{1024\sigma^2}\right)$$

and the result follows when $\rho^* \geqslant 4096\log(2)\sigma/(C_M^{(1)}\sqrt{N})$, which implies that the denominator of the right-hand side is greater than $1/2$.

■

Conclusion: construction of the event Ω_0

We define the event

$$\Omega_0 = \Omega^* \cap \bigcap_{k=0}^{K_0} A_k.$$

It follows from Proposition 3.2.1 and Proposition 3.2.2 (as well as Lemma 3.2.4 below), that, as long as $\rho^* \geqslant 4096\log(2)\sigma/(C_M^{(1)}\sqrt{N})$,

$$\mathbb{P}[\Omega_0] \geqslant 1 - 4\exp(-C_6 N) - 40\exp\left(-\frac{C_6' N r_M(\|t^*\|_1)^2)}{\sigma^2}\right)$$

where C_6 and C_6' are absolute constants.

3.2.2. Deterministic part of the proof

We first start with some few lemmas on the growth of $r_M(\cdot)$ and $r_Q(\cdot)$. We then construct a partition of $\mathbb{R}^d$ depending on the behavior of function $r^2(\cdot)$ (in particular, its concavity for intermediate values is an issue; we solve it thanks to a peeling argument). We then turn to the main deterministic argument showing that $\widehat{t}$ belongs to a ℓ_1^d-ball of radius ρ^* around t^*. The latter holds on the event Ω_0 introduced in Section 3.2.1.

Two Lemmas on the growth of r_M and r_Q

Lemma 3.2.3. *Let $\rho > 0$ and $\phi = 4(C_M^{(2)})^2/(C_M^{(1)})^2$. If $\phi\rho \leqslant \rho_0 \min(1,\eta)$, then for any $\rho' \geqslant \phi\rho$,*

$$r_M^2\left(\rho'\right) > 2r_M^2(\rho) \text{ and } r^2\left(\rho'\right) > 2r^2(\rho).$$

Proof. Since $r_M(\cdot)$, $r_Q(\cdot)$ and $r(\cdot)$ are non-decreasing, we only have to prove the result for $\rho' = \phi\rho$. Recall that $C_M^{(2)} \geqslant C_M^{(1)}$ (so $\phi \geqslant 4$). First note that if $N \geqslant \zeta' d$ then $r_Q(\rho) = 0$ and so the second claim follows from the first one since $r(\rho) = r_M(\rho)$ in this case. And when $N \leqslant \zeta d$, one has $r_Q^2(\phi\rho) = \phi^2 r_Q^2(\rho) \geqslant 16 r_Q^2(\rho) > 2r_Q^2(\rho)$ (because in that case $r_Q(\rho)^2 > 0$). Therefore the second claim is a straightforward consequence of the first one. So it only remains to study the behavior of $r_M^2(\cdot)$. For $\rho < \phi\rho \leqslant \rho_0$, r_M is given by one of the two last expressions of (3.1.9).

First assume that $(\phi\rho)^2 N \leqslant \sigma^2 \log d$ then

$$r_M^2(\phi\rho) \geqslant C_M^{(1)} \phi\rho\sigma\sqrt{\log(ed/N)} > 2C_M^{(2)}\rho\sigma\sqrt{\log(ed/N)} \geqslant 2r_M^2(\rho).$$

Now, assume that $\sigma^2 \log d \leqslant \rho^2 N \leqslant (\phi\rho)^2 N$ then

$$r_M^2(\phi\rho) = C_M \phi\rho\sigma\sqrt{\frac{1}{N}\log\left(\frac{e\sigma d}{\phi\rho\sqrt{N}}\right)}$$

for some $C_M \in [C_M^{(1)}, C_M^{(2)}]$. One has that for all $x \geqslant \phi e$, $\log(x/\phi) > \log(x)/\phi$, and, since $\rho_0 = \sigma d/(\eta\sqrt{N})$, the assumption $\phi\rho \leqslant \rho_0 \min(1,\eta)$ guarantees that $e\sigma d/\rho\sqrt{N} \geqslant \phi e$. Therefore, we have:

$$r_M^2(\phi\rho) > C_M^{(1)}\phi\rho\sigma\sqrt{\frac{1}{\phi N}\log\left(\frac{e\sigma d}{\rho\sqrt{N}}\right)} \geqslant 2C_M^{(2)}\rho\sigma\sqrt{\frac{1}{N}\log\left(\frac{e\sigma d}{\rho\sqrt{N}}\right)} \geqslant 2r_M^2(\rho).$$

Finally, when $\rho^2 N \leqslant \sigma^2 \log d \leqslant (\phi\rho)^2 N$, since r_M is increasing, it is clear considering the two previous cases that one has again $r_M^2(\phi\rho) > (\sqrt{\phi}C_M^{(1)}/C_M^{(2)})r_M^2(\rho)$, so $r_M^2(\phi\rho) > 2r_M^2(\rho)$. ∎

Lemma 3.2.4. *Let $\nu > 0$. If $\nu \geqslant 1$ then $r_M(\nu\rho) \leqslant \sqrt{\nu} r_M(\rho)$ and $r(\nu\rho) \leqslant \nu r(\rho)$. If $\nu \leqslant 1$ then $r_M(\nu\rho) \geqslant \sqrt{\nu} r_M(\rho)$ and $r(\nu\rho) \geqslant \nu r(\rho)$.*

Proof. It is clear that $r_Q(\nu\rho) = \nu r_Q(\rho)$, because $\ell^*(\nu\rho B_1^d \cap \nu r_Q(\rho) B_2^d) = \nu\ell^*(\rho B_1^d \cap r_Q(\rho) B_2^d)$. As for $r_M(\nu\rho)$, for $\nu \geqslant 1$, one has
$\sigma\ell^*(\nu\rho B_1^d \cap \sqrt{\nu} r_M(\rho) B_2^d) \leqslant \sigma\nu\ell^*(\rho B_1^d \cap r_M(\rho) B_2^d) = \nu\eta\sqrt{N} r_M(\rho)^2 = \eta\sqrt{N}(\sqrt{\nu} r_M(\rho))^2$. So $r_M(\nu\rho) \leqslant \sqrt{\nu} r_M(\rho)$.

As for the case $\nu \leqslant 1$, then $1/\nu \geqslant 1$, and it suffices to write that $r_M(\rho) = r_M\left((1/\nu)\nu\rho\right) \leqslant (1/\nu) r_M(\nu\rho)$ to get the result (and still $r_Q(\rho/\nu) = r_Q(\rho)/\nu$).

∎

Partition of $\mathbb{R}^d$ into three zones

Recall that $\rho^* = \max\left(10, 8(C_M^{(2)})^2/((C_M^{(1)})^2\eta) + 1\right)\|t^*\|_1$, that ρ_0 is the smallest ρ such that $r_M(\rho) = r_M(\rho_0)$ for all $\rho \geqslant \rho_0$ and that $K_0 = \min\{k \in \mathbb{N} : 2^k\rho^* \geqslant 2\rho_0\}$.

Hereunder, K_0 is the number of shelves used to partition the intermediate "peeling zone" defined below. We use ρ^* and K_0 to construct a partition of $\mathbb{R}^d$ into three main zones:

- the "central zone" $t^* + \rho^* B_1^d$,
- the intermediate "peeling zone" $\{t \in \mathbb{R}^d : \rho^* < \|t - t^*\|_1 \leqslant 2^{K_0}\rho^*\}$ (recall that $2^{K_0}\rho^* \simeq 2\rho_0$). This zone is considered only when $K_0 \geqslant 1$. We use a "peeling", i.e. a partition of this zone into K_0 sub-areas called the "shelves": $2^{k-1}\rho^* < \|t - t^*\|_1 \leqslant 2^k\rho^*$, for $k = 1, \cdots, K_0$.
- the "exterior zone" $\{t \in \mathbb{R}^d : \|t - t^*\|_1 > 2^{K_0}\rho^*\}$, on which $r_M(\cdot)$ is constant.

Our main objective is now to show that, on the event Ω_0, $\widehat{t}$ belongs to the central zone.

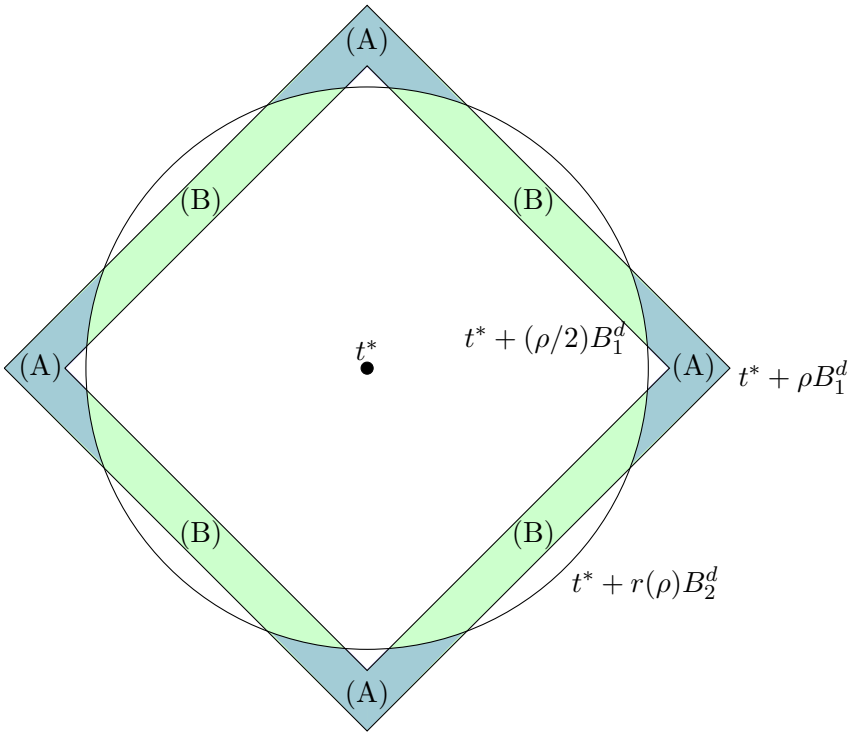


Figure 3.2: $P_N\mathcal{M}_{t-t^*}$ is dominated by $P_N\mathcal{Q}_{t-t^*}$ in (A) and by $\mathcal{R}_{t,t^*}$ in (B). The regularization function $\Psi(\cdot)$ is designed in order to dominate $P_N\mathcal{M}_{t-t^*}$ in (B).

Locating $\widehat{t}$ in the central zone on the event Ω_0

To show that, on the event Ω_0, $\widehat{t}$ belongs to the central zone, it is enough to show that any t outside this area satisfies $P_N\mathcal{L}_t^\Psi > 0$ where we recall that in our case,

$$P_N\mathcal{L}_t^\Psi := \left(\frac{1}{N}\sum_{i=1}^N (Y_i - \langle X_i, t\rangle)^2 + c_0 r^2(\|t\|_1)\right) - \left(\frac{1}{N}\sum_{i=1}^N (Y_i - \langle X_i, t^*\rangle)^2 + c_0 r^2(\|t^*\|_1)\right).$$

This will indeed prove that $\|\widehat{t} - t^*\|_1 \leqslant \rho^*$ since $P_N\mathcal{L}_{\widehat{t}}^\Psi \leqslant 0 = P_N\mathcal{L}_{t^*}^\Psi$.

Let $t \in \mathbb{R}^d$ be outside the central zone – i.e. $\|t - t^*\|_1 > \rho^*$. First note that t satisfies $\|t\|_1 \geqslant \|t-t^*\|_1 - \|t^*\|_1 \geqslant \rho^* - \|t^*\|_1 \geqslant 9\|t^*\|_1$. Therefore, we have $\mathcal{R}_{t,t^*} = \Psi(\|t\|_1) - \Psi(\|t^*\|_1) = c_0 r^2(\|t\|_1) - c_0 r^2(\|t^*\|_1) > 0$ and $(8/9)\|t\|_1 \leqslant \|t - t^*\|_1 \leqslant (10/9)\|t\|_1$.

From now on, we place ourselves on the event Ω_0 introduced in Section 3.2.1.

First assume that $\rho^* < 2\rho_0$, then both the "intermediate peeling zone" and the "exterior zone" must be considered. For t in any of these two areas, one has $\mathcal{R}_{t',t^*} \geqslant 0$.

Let us begin by the "peeling zone". Consider t and $k \geqslant 1$ such that

$$2^{k-1}\rho^* < \|t - t^*\|_1 \leqslant 2^k\rho^* \text{ and } \|t - t^*\|_1 \leqslant 2^{K_0}\rho^*.$$

We recall that $2^{K_0-1}\rho^* < 2\rho_0 \leqslant 2^{K_0}\rho^*$ and ρ_0 is the smallest radius such that $r_M(\rho) = r_M(\rho_0)$ for all $\rho \geqslant \rho_0$.

One possibility is that $\|t - t^*\|_2^2 \geqslant r^2(2^k\rho^*)$, then on Ω^* one has $P_N\mathcal{Q}_{t-t^*} \geqslant \|t - t^*\|_2^2/2$ and, on A_k, one has $|P_N\mathcal{M}_{t-t^*}| \leqslant \|t-t^*\|_2^2/4$. Hence, on Ω_0, $P_N\mathcal{L}_t^\Psi \geqslant \|t-t^*\|_2^2/4 + \mathcal{R}_{t',t^*} > 0$.

Let us tackle now the other case: $\|t-t^*\|_2^2 < r^2(2^k\rho^*)$. We will show that in this situation, if c_0 is large enough, $\mathcal{R}_{t,t^*}/4 > |P_N\mathcal{M}_{t-t^*}|$.

As $\|t\|_1 > 4\|t^*\|_1(C_M^{(2)})^2/(C_M^{(1)})^2$ (t is not in the "central zone") and $4\|t^*\|_1(C_M^{(2)})^2/(C_M^{(1)})^2 \leqslant \eta\rho_0$ (since $2\rho_0 \geqslant \rho^* > 8\|t^*\|_1(C_M^{(2)})^2/((C_M^{(1)})^2\eta)$), by Lemma 3.2.3 one has that $c_0 r^2(\|t\|_1) \geqslant 2c_0 r^2(\|t^*\|_1)$. Thus, $\mathcal{R}_{t,t^*} \geqslant c_0 r^2(\|t\|_1)/2$. So, thanks to Lemma 3.2.4, and since

$$\frac{\|t\|_1}{2^k\rho^*} = \frac{\|t - t^*\|_1}{2^k\rho^*}\frac{\|t\|_1}{\|t - t^*\|_1} > \frac{1}{2}\cdot\frac{9}{10} > \frac{2}{5},$$

one has

$$\mathcal{R}_{t,t^*} \geqslant \frac{1}{2}c_0 r^2(\|t\|_1) \geqslant \frac{1}{2}\cdot\left(\frac{\|t\|_1}{2^k\rho^*}\right)^2 c_0 r^2(2^k\rho^*) > (2/25)c_0 r^2(2^k\rho^*).$$

In addition, we have $\|t - t^*\|_1 \leqslant 2^k\rho^*$ and $\|t - t^*\|_2^2 < r^2(2^k\rho^*)$, hence, on the event A_k, $|P_N\mathcal{M}_{t-t^*}| \leqslant r^2(2^k\rho^*)/4$. As a consequence, for $c_0 > 13$ one has

$$\frac{\mathcal{R}_{t,t^*}}{4} - |P_N\mathcal{M}_{t-t^*}| > \frac{2}{25}\cdot c_0 r^2(2^k\rho^*)\cdot\frac{1}{4} - \frac{1}{4}r^2(2^k\rho^*) > 0. \tag{3.2.10}$$

A fortiori, $\mathcal{R}_{t',t^*} - |P_N\mathcal{M}_{t-t^*}| > 0$ and as $P_N\mathcal{Q}_{t-t^*} \geqslant 0$, this implies in particular that $P_N\mathcal{L}_t^\Psi > 0$.

To sum up, we proved that for all t in the peeling zone, we have

$$P_N\mathcal{L}_t^\Psi \geqslant P_N\mathcal{Q}_{t-t^*} - |P_N\mathcal{M}_{t-t^*}| + \mathcal{R}_{t',t^*}/4 > 0. \tag{3.2.11}$$

Let us now study the exterior zone. We will mainly use homogeneity arguments. Let $t \in \mathbb{R}^d$ be outside the ball $t^* + 2^{K_0}\rho^* B_1^d$. Define $t' \in t^* + 2^{K_0}\rho^* S_1^{d-1}$ such that $t - t^* = \alpha_t(t' - t^*)$ for some $\alpha_t \geqslant 1$. In particular, $\|t' - t^*\|_2 \geqslant 2r_0 > r_M(2^{K_0}\rho^*)$ so by Proposition 3.2.2, on the event A_{K_0}, one has $|P_N\mathcal{M}_{t'-t^*}| \leqslant \|t' - t^*\|_2^2/4$. We consider now two cases.

If $\|t' - t^*\|_2 \geqslant r_Q(\|t' - t^*\|_1)$, then by Lemma 3.2.1, on the event Ω^*, one has $P_N\mathcal{Q}_{t'-t^*} \geqslant \|t' - t^*\|_2^2/2$. So on the event Ω_0, $P_N\mathcal{Q}_{t'-t^*} - |P_N\mathcal{M}_{t'-t^*}| > 0$ and

$$P_N\mathcal{Q}_{t-t^*} - |P_N\mathcal{M}_{t-t^*}| = \alpha_t^2 P_N\mathcal{Q}_{t'-t^*} - \alpha_t|P_N\mathcal{M}_{t'-t^*}| \geqslant \alpha_t\left(P_N\mathcal{Q}_{t'-t^*} - |P_N\mathcal{M}_{t'-t^*}|\right) > 0$$

therefore $P_N\mathcal{L}_t^\Psi > 0$.

On the contrary, if $\|t' - t^*\|_2 < r_Q(\|t' - t^*\|_1)$, as $\|t' - t^*\|_2 \geqslant 2r_0$, then $r_Q(\|t'\|_1) \geqslant 9r_Q(\|t' - t^*\|_1)/10 > 9r_0/5 = 9r_M(\|t'\|_1)/5 > r_M(\|t'\|_1)$ so $r(\|t'\|_1) = r_Q(\|t'\|_1)$. One has $\|t\|_1 = \|t^* + \alpha_t(t' - t^*)\|_1 \geqslant (4\alpha_t/5)\|t'\|_1$ since $\alpha_t \geqslant 1$ and $\|t' - t^*\|_1 \geqslant \rho^*$. So $r^2(\|t\|_1) \geqslant r_Q^2\left(\alpha_t 4\|t'\|_1/5\right) = 4^2\alpha_t^2 r^2(\|t'\|_1)/5^2$. We have seen before that $r_Q(\|t'\|_1) \geqslant 9r_0/5 \geqslant 9r_M(\|t^*\|_1)/5$, and $r_Q(\|t'\|_1) \geqslant 9r_Q(\|t^*\|_1)$, so $r^2(\|t'\|_1) \geqslant 3r^2(\|t^*\|_1)$. As a consequence,

$$\begin{aligned}\mathcal{R}_{t,t^*} &= c_0 r^2(\|t\|_1) - c_0 r^2(\|t^*\|_1) \geqslant c_0\left(\alpha_t^2\frac{4^2}{5^2}r^2(\|t'\|_1) - (1/3)r^2(\|t'\|_1)\right)\\ &\geqslant \alpha_t\frac{c_0}{4}r^2(\|t'\|_1) \geqslant \alpha_t\mathcal{R}_{t',t^*}/4.\end{aligned}$$

Moreover, since $t - t^* = \alpha_t(t' - t^*)$, one has $P_N\mathcal{Q}_{t-t^*} = \alpha_t^2 P_N\mathcal{Q}_{t'-t^*}$ and $P_N\mathcal{M}_{t-t^*} = \alpha_t P_N\mathcal{M}_{t'-t^*}$. So, in the case $\|t' - t^*\|_2 < r_Q(\|t' - t^*\|_1)$, by (3.2.11) applied to t',

$$P_N\mathcal{L}_t^\Psi \geqslant \alpha_t(P_N\mathcal{Q}_{t'-t^*} - |P_N\mathcal{M}_{t'-t^*}| + \mathcal{R}_{t',t^*}/4) > 0.$$

Let us now consider the case $\rho^* > 2\rho_0$. In this situation, there is no need for the intermediate "peeling" zone. Let $t \in \mathbb{R}^d$ be outside the ball $t^* + \rho^* B_1^d$ and set $t' \in t^* + \rho^* S_1^{d-1}$ such that $t - t^* = \alpha_t(t' - t^*)$ with $\alpha_t \geqslant 1$. Then one can apply arguments similar to the peeling case (with $\|t' - t^*\|_1 = \rho^*$, but this time $k = K_0 = 0$), on the event $A_0 \cap \Omega^*$, to t'. If $\|t' - t^*\|_2 \geqslant r_Q(\|t' - t^*\|_1)$, then by Proposition 3.2.1, on the event Ω_0 one has $P_N\mathcal{Q}_{t'-t^*} \geqslant |P_N\mathcal{M}_{t'-t^*}|/2$. Conversely, if $\|t' - t^*\|_2 < r_Q(\|t' - t^*\|_1)$, then $r^2(\|t'\|_1) \geqslant 3r^2(\|t^*\|_1)$ so for c_0 big enough, in the same spirit as (3.2.10), one gets that $\mathcal{R}_{t',t^*}/4 - |P_N\mathcal{M}_{t'-t^*}| > 0$. So in both cases, $P_N\mathcal{Q}_{t-t^*} - |P_N\mathcal{M}_{t-t^*}| + \mathcal{R}_{t',t^*}/4 > 0$. The same argument as previously for the exterior zone, shows that $P_N\mathcal{L}_t^\Psi \geqslant \alpha_t(P_N\mathcal{Q}_{t'-t^*} - |P_N\mathcal{M}_{t'-t^*}| + \mathcal{R}_{t',t^*}/4)$, so $P_N\mathcal{L}_t^\Psi > 0$. As a conclusion, in the case $\rho^* > 2\rho_0$, we have for all $t \in \mathbb{R}^d$ satisfying $\|t - t^*\|_1 \geqslant \rho^* > 2\rho$ that $P_N\mathcal{L}_t^\Psi > 0$.

To sum up, on the event Ω_0, any t outside the central zone satisfies $P_N\mathcal{L}_t^\Psi > 0$. Therefore, given that $\widehat{t}$ satisfies $P_N\mathcal{L}_{\widehat{t}}^\Psi \leqslant 0$, we conclude that $\widehat{t}$ belongs to the central zone.

Conclusion of the proof of Theorem 3.1.4

On the event Ω_0, $\widehat{t} \in (t^* + \rho^* B_1^d)$. Hence, either $\|\widehat{t} - t^*\|_2^2 \leqslant r^2(\rho^*)$ and the proof is over or $\|\widehat{t} - t^*\|_2^2 > r^2(\rho^*)$. In the latter case, one has

$$P_N\mathcal{Q}_{\widehat{t}-t^*} > \frac{1}{2}\|\widehat{t} - t^*\|_2^2 \text{ and } |P_N\mathcal{M}_{\widehat{t}-t^*}| < \frac{1}{4}\|\widehat{t} - t^*\|_2^2$$

and so

$$0 \geqslant P_N \mathcal{L}_{\widehat{t}}^{\Psi} \geqslant \frac{1}{4}\|\widehat{t} - t^*\|_2^2 + c_0 r^2(\|\widehat{t}\|_1) - c_0 r^2(\|t^*\|_1)$$

which implies $\|\widehat{t} - t^*\|_2^2 \leqslant 4c_0 r^2(\|t^*\|_1)$.

Thus, taking $\theta_0 = \max\left(100, (8(C_M^{(2)})^2/(C_M^{(1)})^2 + 1)^2, 4c_0\right)$, with probability at least

$$\mathbb{P}[\Omega_0] \geqslant 1 - 4\exp\left(-C_6 N\right) - 40\exp\left(-C_6' N r_M(\|t^*\|_1)^2/\sigma^2\right)$$

one gets that in both cases, $R(\widehat{t}) - R(t^*) = \|\widehat{t} - t^*\|_2^2 \leqslant \theta_0 r^2(\|t^*\|_1)$. Moreover, for $\|t^*\|_1 \geqslant \Delta_0 \sigma\sqrt{\log(ed)/N}$ for Δ_0 an absolute constant large enough, we have $\mathbb{P}[\Omega_0] \geqslant 3/4$. Given that $r^2(\|t^*\|_1)$ is the minimax rate of convergence over $\|t^*\|_1 B_1^d$ (cf. Theorem 3.1.3), we conclude that $\Psi(\rho) = c_0 r^2(\rho)$ is indeed a minimax regularization function.

3.3. Technical material and proof of Proposition 3.1.6

3.3.1. Localization with balls and spheres

The next lemma shows that when the intersection is not trivial, localizing by intersecting an ℓ_1 ball with an ℓ_2 sphere or the corresponding full ℓ_2 ball is equivalent.

Lemma 3.3.1. *If $\rho \geqslant r$, then $\ell^*(\rho B_1^d \cap r S_2^{d-1}) = \ell^*(\rho B_1^d \cap r B_2^d)$. If $\rho < r$, then $\ell^*(\rho B_1^d \cap r B_2^d) = \ell^*(\rho B_1^d)$ and $\ell^*(\rho B_1^d \cap r S_2^{d-1}) = 0$.*

Proof. Since for any set $T \subset \mathbb{R}^d$, $\ell^*(T) = \ell^*(\mathrm{conv}(T))$ (with $\mathrm{conv}(T)$ denoting the convex hull of T), the result is a direct consequence of the fact that for $\rho < r$, $\mathrm{conv}(\rho B_1^d \cap r B_2^d) = \rho B_1^d$ and $\mathrm{conv}(\rho B_1^d \cap r S_2^{d-1}) = \emptyset$ (which are two obvious statements), and that if $\rho \geqslant r$, then $\mathrm{conv}(\rho B_1^d \cap r S_2^{d-1}) = \mathrm{conv}(\rho B_1^d \cap r B_2^d)$, which we prove now.

One inclusion is immediate, it remains to prove that $\mathrm{conv}(\rho B_1^d \cap r B_2^d) \subset \mathrm{conv}(\rho B_1^d \cap r S_2^{d-1})$. First,

$$\mathrm{conv}(\rho B_1^d \cap r B_2^d) = \rho B_1^d \cap r B_2^d = \mathrm{conv}\{t \in \mathbb{R}^d : \ \max(\|t\|_1/\rho, \|t\|_2/r) = 1\}$$

so it only remains to show that $\{t \in \mathbb{R}^d : \ \max(\|t\|_1/\rho, \|t\|_2/r) = 1\} \subset \mathrm{conv}(\rho B_1^d \cap r S_2^{d-1})$.

First, remark that $\{t \in \mathbb{R}^d : \|t\|_1/\rho \leqslant 1, \|t\|_2/r = 1\}$ is included in $\rho B_1^d \cap r S_2^{d-1}$. Let us now consider the set $\{t \in \mathbb{R}^d : \|t\|_1/\rho = 1, \|t\|_2/r < 1\}$ and consider an element t in it. We denote by $e_1, \ldots, e_d$ the canonical basis of $\mathbb{R}^d$ and we recall that each face of ρB_1^d is the convex hull of its vertices. So, since $t \in \rho S_1^{d-1}$, there exist $b_1 \in \{\rho e_1, -\rho e_1\}, b_2 \in \{\rho e_2, -\rho e_2\}, \ldots, b_d \in \{\rho e_d, -\rho e_d\}$ such that t is in the convex hull of $\{b_1 \ldots, b_d\}$: there exist non-negative coefficients $\mu_1, \ldots, \mu_d$ such that $\sum_{j=1}^d \mu_j(b_j - t) = 0$. Consider for each $j \in \{1, \ldots, d\}$ the mapping

$$f_j : x \in [0,1] \mapsto \|t + x(b_j - t)\|_2/r - \|t + x(b_j - t)\|_1/\rho$$

This mapping is continuous, $f_j(0) < 0$ (t is in ρS_1^{d-1} and $r B_2^d$ but not in $r S_2^{d-1}$) , and $f_j(1) \geqslant 0$ (because $\|b_j\|_2 = \|b_j\|_1 = \rho$ and $\rho \geqslant r$). So there exists x_j in (0,1] such that $f(x_j) = 0$, which means that $t + x_j(b_j - t) \in \rho S_1^{d-1} \cap r S_2^{d-1}$. One has that

$$\sum_{j=1}^d \frac{\mu_j}{x_j}\left(t + x_j(b_i - t) - t\right) = \sum_{j=1}^d \frac{\mu_j}{x_j} x_j(b_j - t) = \sum_{j=1}^d \mu_j(b_j - t) = 0.$$

As a consequence, t is in the convex hull of the vectors $t + x_j(b_j - t) \in \rho S_1^{d-1} \cap r S_2^{d-1}, j \in \{1, \dots, d\}$ which achieves the proof. ∎

3.3.2. Control of the probability estimate

Lemma 3.3.2. *Set $\eta = 1/(16\sqrt{2})$. For all $\nu > 0$, we have*

$$\sum_{k=0}^{K_0} \exp\left(-\nu r_M\left(2^k\rho^*\right)^2\right) \leqslant \frac{10}{1 - \exp\left(-\nu C_M^{(1)} \frac{\sigma}{4\sqrt{N}}\rho^*\right)} \exp\left(-\frac{C_M^{(1)}}{C_M^{(2)}} \nu r_M\left(\rho^*\right)^2\right).$$

Proof. First, the terms of the sum are non-increasing (remember that r_M is a non-decreasing function). So skipping the last terms will not change the order of magnitude:

$$\sum_{k=0}^{K_0} \exp\left(-\nu r_M\left(2^k\rho^*\right)^2\right) \leqslant \max\left(10\exp\left(-\nu r_M\left(\rho^*\right)^2\right), 10\sum_{k=0}^{K_0-9} \exp\left(-\nu r_M\left(2^k\rho^*\right)^2\right)\right).$$

One has $10\exp\left(-\nu r_M\left(\rho^*\right)^2\right) > 10\sum_{k=0}^{K_0-9} \exp\left(-\nu r_M\left(2^k\rho^*\right)^2\right)$ when $K_0 \leqslant 8$. Let us now assume that $K_0 \geqslant 9$. We study the sum $\sum_{k=0}^{K_0-9} \exp\left(-\nu r_M\left(2^k\rho^*\right)^2\right)$.

In order to get rid of the "range" $[C_M^{(1)}, C_M^{(2)}]$ in the definition of $r_M(\cdot)$, notice that $\sum_{k=0}^{K_0-9} \exp\left(-\nu r_M\left(2^k\rho^*\right)^2\right) \leqslant \sum_{k=0}^{K_0-9} a_k$ with:

$$a_k := \begin{cases} \exp\left(-\nu C_M^{(1)} 2^k\rho^*\sigma\sqrt{\frac{\log(ed)}{N}}\right) & \text{if } \left(2^k\rho^*\right)^2 N \leqslant \sigma^2\log(d) \\ \exp\left(-\nu C_M^{(1)} 2^k\rho^*\sigma\sqrt{\frac{1}{N}\log\left(\frac{e\sigma d}{2^k\rho^*\sqrt{N}}\right)}\right) & \text{otherwise.} \end{cases}$$

We emphasize that the sum goes only up to $2^{K_0}\rho^*$, which excludes the constant third form of $r_M(\cdot)$.

Applying a second time the "range" $[C_M^{(1)}, C_M^{(2)}]$ in the bounds on r_M allows to bound a_0 in terms of $r_M\left(\rho^*\right)^2$: $a_0 \leqslant \exp\left(-(C_M^{(1)}/C_M^{(2)})\nu r_M\left(\rho^*\right)^2\right)$. Therefore, in the following we will bound $\sum_{k=0}^{K_0-9} a_k$ with respect to a_0, and then get back to $r_M\left(\rho^*\right)^2$.

We now prove that there exists α independent on k (but dependent on the other parameters) such that for any $k \leqslant K_0 - 9$, $a_{k+1}/a_k \leqslant \alpha < 1$.

If $(2^{k+1}\rho^*)^2 \leqslant \sigma^2\log(d)/N$ then

$$\begin{aligned} \frac{a_{k+1}}{a_k} &= \exp\left(-\nu C_M^{(1)} 2^{k+1}\rho^*\sigma\sqrt{\frac{\log(ed)}{N}} + \nu C_M^{(1)} 2^k\rho^*\sigma\sqrt{\frac{\log(ed)}{N}}\right) \\ &= \exp\left(-\nu C_M^{(1)} 2^k\rho^*\sigma\sqrt{\frac{\log(ed)}{N}}\right) \leqslant \exp(-\nu C_M^{(1)}\rho^*\beta_1) \end{aligned}$$

with $\beta_1 = \sigma\sqrt{\log(ed)/N} \geqslant \sigma/\sqrt{N}$.

As for the case $(2^k\rho^*)^2 \leqslant \sigma^2\log(d)/N < (2^{k+1}\rho^*)^2$ (which can only occur when $d > 1$), then one has $2^{k+1}\rho^* \leqslant 2\sigma\sqrt{\log d}/\sqrt{N}$ and so

$$\begin{aligned}\frac{a_{k+1}}{a_k} &= \exp\left(-\nu C_M^{(1)} 2^{k+1}\rho^*\sigma\sqrt{\frac{1}{N}\log\left(\frac{e\sigma d}{2^{k+1}\rho^*\sqrt{N}}\right)} + \nu C_M^{(1)} 2^k\rho^*\sigma\sqrt{\frac{\log(ed)}{N}}\right)\\ &\leqslant \exp\left(-\nu C_M^{(1)} 2^k\rho^*\frac{\sigma}{\sqrt{N}}\left(2\sqrt{\log\left(\frac{ed}{2\sqrt{\log d}}\right)} - \sqrt{\log ed}\right)\right)\\ &\leqslant \exp\left(-\nu C_M^{(1)} 2^k\rho^*\frac{\sigma}{2\sqrt{N}}\right) \leqslant \exp\left(-\nu C_M^{(1)}\rho^*\frac{\sigma}{2\sqrt{N}}\right)\end{aligned}$$

The second inequality relies on a straightforward analysis fact:

$$\forall\ d \geqslant 2, \quad \left(2\sqrt{\log\left(\frac{ed}{2\sqrt{\log d}}\right)} - \sqrt{\log ed}\right) > \frac{1}{2}$$

Let us tackle now the case $(2^k\rho^*)^2 > \sigma^2\log(d)/N$. We have $a_{k+1}/a_k = \exp(b_k\nu C_M^{(1)} 2^k\rho^*\sigma/\sqrt{N})$ where

$$b_k := -2\sqrt{\log\left(\frac{e\sigma d}{2^{k+1}\rho^*\sqrt{N}}\right)} + \sqrt{\log\left(\frac{e\sigma d}{2^{k+1}\rho^*\sqrt{N}}\right) + \log(2)}.$$

Since $\sqrt{x+y} \leqslant \sqrt{x} + \sqrt{y}$ for all $x, y \geqslant 0$, one has (still for $k \leqslant K_0 - 9$):

$$b_k \leqslant \sqrt{\log(2)} - \sqrt{\log\left(\frac{e\sigma d}{2^{k+1}\rho^*\sqrt{N}}\right)} \leqslant \sqrt{\log(2)} - \sqrt{\log\left(\frac{e\sigma d}{2^{K_0-8}\rho^*\sqrt{N}}\right)} \leqslant (1-\sqrt{2})\sqrt{\log(2)}$$

because $2^{K_0-8}\rho^* \leqslant 2^{-6}\rho_0 \leqslant 2^{-6}\sigma d/(\eta\sqrt{N}) \leqslant \sigma d/(2\sqrt{N})$ (we recall that $\rho_0 = \sigma d/(\eta\sqrt{N})$ and $\eta = 1/(16\sqrt{2})$). It follows that

$$\frac{a_{k+1}}{a_k} = \exp\left(\nu C_M^{(1)} 2^k\rho^*\frac{\sigma}{\sqrt{N}}b_k\right) \leqslant \exp\left(-\nu C_M^{(1)}\rho^*\frac{\sigma}{\sqrt{N}}(-b_k)\right) \leqslant \exp(-\nu C_M^{(1)}\rho^*\beta_2)$$

with $\beta_2 = (\sqrt{2}-1)\sqrt{\log(2)}\sigma/\sqrt{N} \geqslant \sigma/(4\sqrt{N})$. Then, we conclude that for all $k \leqslant K_0 - 9$, $a_{k+1}/a_k \leqslant \exp\left(-\nu C_M^{(1)}\frac{\sigma}{4\sqrt{N}}\rho^*\right)$ and so

$$\sum_{k=0}^{K_0-9} a_k \leqslant a_0 \sum_{k=0}^{K_0-9} \exp\left(-k\nu C_M^{(1)}\frac{\sigma}{4\sqrt{N}}\rho^*\right) = a_0\frac{1-\exp\left(-(K_0-8)\nu C_M^{(1)}\frac{\sigma}{4\sqrt{N}}\rho^*\right)}{1-\exp\left(-\nu C_M^{(1)}\frac{\sigma}{4\sqrt{N}}\rho^*\right)}.$$

Finally, the result follows, since $a_0 \leqslant \exp\left(-(C_M^{(1)}/C_M^{(2)})\nu r_M(\rho^*)^2\right)$ and

$$\frac{1-\exp\left(-(K_0-8)\nu C_M^{(1)}\frac{\sigma}{4\sqrt{N}}\rho^*\right)}{1-\exp\left(-\nu C_M^{(1)}\frac{\sigma}{4\sqrt{N}}\rho^*\right)} \leqslant \frac{1}{1-\exp\left(-\nu C_M^{(1)}\frac{\sigma}{4\sqrt{N}}\rho^*\right)}.$$

■

3.3.3. Proof of Proposition 3.1.6

The proof of Proposition 3.1.6 relies on key ideas developed in minimax theory. We refer the reader to Tsybakov [2009] for a state of the art in minimax theory.

Let $\widehat{t}$ satisfy the properties of Proposition 3.1.6, i.e. a procedure adaptive on the finite set $\{\pm\rho e_1, \cdots, \pm\rho e_d\}$ and let $t^* \in (\rho/2)B_1^d$. Denote $\Lambda = \{t^*, \pm\rho e_1, \ldots, \pm\rho e_d\} = \{t_0^*, t_1^*, \cdots, t_{2d}^*\}$ so that $t_0^* = t^*$. It is straightforward to check that Λ is a $\rho/2$-separated set in ρB_1^d w.r.t. ℓ_2^d.

Now, let us define the test statistics

$$\widehat{\phi} \in \operatorname*{argmin}_{j\in\{0,\cdots,2d\}} \|t_j^* - \widehat{t}\|_2.$$

One has for all $j \in \{0, 1, \ldots, 2d\}$ that, if $\widehat{\phi} \neq j$ then $\|\widehat{t} - t_j^*\|_2 \geqslant \rho/4$. Indeed, if $\widehat{\phi} \neq j$ then there exists $k \in \{0, 1, \ldots, 2d\}\backslash\{j\}$ such that $\|t_k^* - \widehat{t}\|_2 \leqslant \|t_j^* - \widehat{t}\|_2$. If $\|t_k^* - \widehat{t}\|_2 \geqslant \rho/4$ then the result holds otherwise $\|t_k^* - \widehat{t}\|_2 < \rho/4$ and so $\|t_j^* - \widehat{t}\|_2 \geqslant \|t_j^* - t_k^*\|_2 - \|t_k^* - \widehat{t}\|_2 \geqslant \rho/4$. Therefore, we have, for all $\tau > 0$

$$\begin{aligned}
\mathbb{P}_{t_0^*}\left[\|\widehat{t} - t_0^*\|_2 \geqslant \rho/4\right] &\geqslant \mathbb{P}_{t_0^*}\left[\widehat{\phi} \neq 0\right] = \sum_{j=1}^{2d} \mathbb{P}_{t_0^*}\left[\widehat{\phi} = j\right] \geqslant \sum_{j=1}^{2d} \tau \mathbb{P}_{t_j^*}\left[\widehat{\phi} = j \text{ and } \frac{d\mathbb{P}_{t_0^*}}{d\mathbb{P}_{t_j^*}} \geqslant \tau\right] \\
&\geqslant \tau \sum_{j=1}^{2d} \mathbb{P}_{t_j^*}\left[\widehat{\phi} = j\right] - \mathbb{P}_{t_j^*}\left[\frac{d\mathbb{P}_{t_0^*}}{d\mathbb{P}_{t_j^*}} < \tau\right] \geqslant \tau \sum_{j=1}^{2d} \mathbb{P}_{t_j^*}\left[\|\widehat{t} - t_j^*\|_2 < \rho/4\right] - \mathbb{P}_{t_j^*}\left[\frac{d\mathbb{P}_{t_0^*}}{d\mathbb{P}_{t_j^*}} < \tau\right].
\end{aligned} \tag{3.3.1}$$

It follows from the adaptation property of $\widehat{t}$ over $\{\pm\rho e_1, \cdots, \pm\rho e_d\}$ that for every $j \in \{1, \cdots, 2d\}$,

$$\mathbb{P}_{t_j^*}\left[\|\widehat{t} - t_j^*\|_2 < \rho/4\right] \geqslant \mathbb{P}_{t_j^*}\left[\|\widehat{t} - t_j^*\|_2^2 < \chi_1 r^2(\rho)\right] \geqslant 3/4 \tag{3.3.2}$$

when $\chi_1 r^2(\rho) \leqslant \rho^2/16$. Let $j \in \{1, \cdots, 2d\}$. Following the same argument as in Proposition 2.3 in Tsybakov [2009] (based on second Pinsker inequality), we obtain

$$\begin{aligned}
\mathbb{P}_{t_j^*}\left[\frac{d\mathbb{P}_{t_0^*}}{d\mathbb{P}_{t_j^*}} \geqslant \tau\right] &= \mathbb{P}_{t_j^*}\left[\frac{d\mathbb{P}_{t_j^*}}{d\mathbb{P}_{t_0^*}} \leqslant \frac{1}{\tau}\right] = 1 - \mathbb{P}_{t_j^*}\left[\log\frac{d\mathbb{P}_{t_j^*}}{d\mathbb{P}_{t_0^*}} > \log(1/\tau)\right] \\
&\geqslant 1 - \frac{1}{\log(1/\tau)} \int \left(\log\frac{d\mathbb{P}_{t_j^*}}{d\mathbb{P}_{t_0^*}}\right)_+ d\mathbb{P}_{t_j^*} \geqslant 1 - \frac{1}{\log(1/\tau)}\left[K(\mathbb{P}_{t_j^*}, \mathbb{P}_{t_0^*}) + \sqrt{2K(\mathbb{P}_{t_j^*}, \mathbb{P}_{t_0^*})}\right]
\end{aligned}$$

where $K(\mathbb{P}_{t_j^*}, \mathbb{P}_{t_0^*})$ denotes the Kullback-Leibler divergence between $\mathbb{P}_{t_j^*}$ and $\mathbb{P}_{t_0^*}$. Since the noise is Gaussian and independent of X in (3.1.7) and X is isotropic, we have $K(\mathbb{P}_{t_j^*}, \mathbb{P}_{t_0^*}) = N\|t_j^* - t_0^*\|_2^2/(2\sigma^2) \leqslant N9\rho^2/(8\sigma^2)$. Hence, if $\rho \leqslant 2\sigma\sqrt{\log(1/\tau)/(96N)}$ and $\log(1/\tau) \geqslant 544/225$, one has

$$\mathbb{P}_{t_j^*}\left[\frac{d\mathbb{P}_{t_0^*}}{d\mathbb{P}_{t_j^*}} \geqslant \tau\right] \geqslant \frac{3}{4}.$$

Using the latter result together with (3.3.2) in (3.3.1) for the values $\tau = 1/(2d)$, we obtain that

$$\mathbb{P}_{t_0^*}\left[\|\widehat{t} - t_0^*\|_2 \geqslant \rho/4\right] \geqslant \frac{1}{2}.$$

■

3.4. Minimax regularization function in the fixed design setup

In this section, we consider the Gaussian regression model with fixed design. We are therefore given a deterministic $N \times d$ design matrix $\mathbb{X}$, whose i-th row vector is denoted by $X_i \in \mathbb{R}^d$, and we observe N noisy projections of a vector $t^* \in \mathbb{R}^d$, $Y_i = \langle X_i, t^* \rangle + \xi_i, i = 1, \dots, N$, where the noises $\xi_1, \dots, \xi_N$ are independent centered Gaussian variables with variances σ^2. The data $(Y_i, X_i)_{i=1}^N$ are used to construct estimators of t^* and the only difference with the previous random design setup is that the X_i's are deterministic vectors whereas so far they were random vectors. We will use most of the time the matrix form of the data: $Y = \mathbb{X}t + \xi$ where $Y = (Y_i)_{i=1}^N$ and $\xi \sim \mathcal{N}(0, \sigma^2 I_N)$ where I_N is the $N \times N$ identity matrix. Note that the fixed design setup is usually considered in signal processing over a finite grid or in experiences where the statistician can chose in advance the design of an experience and then observed an output.

In order to design a minimax regularization function in this setup, we need to adapt the definitions introduced in Section 3.1 to the fixed design case. We first start with the definition of a minimax rate of convergence over a subset of $\mathbb{R}^d$. We use the empirical (or normalized) Euclidean inner product and norm: for all $u, v \in \mathbb{R}^N$,

$$\langle u, v \rangle_{L_2^N} = \frac{1}{N}\sum_{i=1}^N u_i v_i \text{ and } \|u\|_{L_N^2} = \sqrt{\frac{1}{N}\sum_{i=1}^N u_i^2}$$

and the associated unit ball $B_{L_N^2} = \{u \in \mathbb{R}^N : \|u\|_{L_N^2} \leqslant 1\}$.

Definition 3.4.1. *Let $T \subset \mathbb{R}^d$, $\mathbb{X}$ denote a (deterministic) $N \times d$ design matrix and ξ be a centered Gaussian random vector in $\mathbb{R}^N$ with covariance matrix $\sigma^2 I_N$. For all $t^* \in T$, define the random vector $Y^{t^*} = \mathbb{X}t^* + \xi$ and denote by $\mathcal{Y}^T := \{Y^{t^*} : t^* \in T\}$ the set of all such random vectors.*

Let $\widehat{t}_N$ be a statistics from $\mathbb{R}^N$ to $\mathbb{R}^d$. Let $0 < \delta_N < 1$ and $\zeta_N > 0$. We say that $\widehat{t}_N$ ***performs with accuracy*** *ζ_N* ***and confidence*** *$1 - \delta_N$* ***relative to the set of targets*** *$\mathcal{Y}^T$, if for all $t^* \in T$, with probability, w.r.t. to a vector Y distributed as Y^{t^*}, at least $1 - \delta_N$, $\|\mathbb{X}(\widehat{t}_N(Y) - t^*)\|_{L_N^2}^2 \leqslant \zeta_N$.*

We say that $\mathcal{R}_N$ ***is a minimax rate of convergence over*** *T* ***for the confidence*** *$1 - \delta_N$ if the two following hold:*

1. *there exists a statistics $\widehat{t}_N$ which performs with accuracy $\mathcal{R}_N$ and confidence $1 - \delta_N$ relative to the set of targets $\mathcal{Y}^T$*

2. *there exists an absolute constant* $g_0' > 0$ *such that if* $\tilde{t}_N$ *is a statistics which performs with accuracy* ζ_N *and confidence* $1-\delta_N$ *relative to the set of targets* $\mathcal{Y}^T$ *then necessarily* $\zeta_N \geqslant g_0' \mathcal{R}_N$.

Note that we use the empirical L_2^N-metric $\|\mathbb{X}\cdot\|_{L_N^2}$ (to the square) with respect to the design $\mathbb{X}$ as a measure of performances of estimators in Definition 3.4.1. The reason we do so is that it is the natural counterpart to the random design case – that is when $\mathbb{X}$ is a standard Gaussian matrix then $\mathbb{E}\|\mathbb{X}t\|_{L_N^2}^2 = \|t\|_2^2$ and the ℓ_2^d-norm is the metric used to measure the performance of estimators in the random design setup – and that it is the natural metric associated to the prediction of Y problem given that if $R(t) = \mathbb{E}\|Y - \mathbb{X}t\|_{L_N^2}^2$ is the risk of t for all $t \in \mathbb{R}^d$ then we have for any estimator $\widehat{t}_N$, $R(\widehat{t}_N) = R(t^*) + \|\mathbb{X}(\widehat{t}_N - t^*)\|_{L_N^2}^2$. So that predicting Y is the same problem as estimating t^* with respect to the empirical $\|\mathbb{X}\cdot\|_{L_N^2}$ metric.

Now, we adapt the definition of a minimax regularization function to the fixed design setup in the next definition.

Definition 3.4.2. *Let* $\|\cdot\|$ *be a norm on* $\mathbb{R}^d$, $T \subset \mathbb{R}^d$ *and* $0 < \delta_N < 1$. *Let us consider the following RERM for some given function* $\Psi : \mathbb{R}_+ \to \mathbb{R}$*:*

$$\widehat{t} \in \operatorname*{argmin}_{t\in\mathbb{R}^d} \left(\|Y - \mathbb{X}t\|_{L_N^2}^2 + \Psi(\|t\|) \right)$$

constructed from a $N \times d$ *deterministic matrix* $\mathbb{X}$ *and a random vector* $Y = \mathbb{X}t^* + \xi$, *with* $\xi \sim \sigma\mathcal{N}(0, I_N)$. *We say that* Ψ *is a* ***minimax regularization function for the norm*** $\|\cdot\|$ ***over*** T ***for the confidence*** $1-\delta_N$, *if there exists an absolute constant* $g_1' > 0$ *such that for all* $t^* \in T$, *the RERM* $\widehat{t}$ *is such that with probability at least* $1-\delta_N$, $\|\widehat{t} - t^*\|_{L_N^2}^2 \leqslant g_1' \mathcal{R}_N$, *where* $\mathcal{R}_N$ *is the minimax rate of convergence over* $\{t \in \mathbb{R}^d : \|t\| \leqslant \|t^*\|\}$.

The statistical bounds one can prove in the fixed design setup depend generally on the property of the design matrix $\mathbb{X}$. Many different assumptions have been introduced during the last two decades in high-dimensional statistics and we refer to Van De Geer and Bühlmann [2009] for some of them. In particular, norm preserving properties like the RIP or weaker assumption on the restricted eigenvalues like the REC or CC have played an important role in statistics (cf. Bickel et al. [2009], Bühlmann and van de Geer [2011], Candès and Tao [2010], Giraud [2014], van de Geer [2007], Van De Geer and Bühlmann [2009]). In this chapter, we assume that $\mathbb{X}$ satisfies the "Restricted Isometry Property". It appears that this condition is equivalent (up to constants) to the property satisfied by a standard Gaussian matrix as in Lemma 3.2.1.

Assumption 3.4.1 (RIP(s))**.** *If* $N < d$ *and* $N/\log(ed/N) > 1$ *then we set* $s = N/\log(ed/N)$. *We assume that all* t *in* $\Sigma_s := \{x \in \mathbb{R}^d : \|x\|_0 \leqslant s\}$ *is such that*

$$\frac{1}{2}\|t\|_2 \leqslant \|\mathbb{X}t\|_{L_N^2} \leqslant \frac{3}{2}\|t\|_2 \tag{3.4.1}$$

where $\|x\|_0$ *is the size of the support of* x. *If* $N \geqslant d$, *we assume that* (3.4.1) *is satisfied for all* $t \in \mathbb{R}^d$.

Note that in the high-dimensional case, i.e. $d > N$, only the situation $N/\log(ed/N) > 1$ is considered in Assumption 3.4.1 to avoid the ultra-high dimensional phenomena discovered in Verzelen [2012]. RIP was introduced in Candes and Tao [2005] and it has been widely used and discussed (cf. for example Davenport and Wakin [2010], Baraniuk et al. [2008] or Garg and Khandekar [2009]), in particular in the field of Compressed Sensing. From our perspective, we use this result for two reasons:

- 1) the minimax results over ℓ_1^d balls we need to develop for our proof of minimax regularization function has been obtained in the fixed design under this condition (or an equivalent one) in Rigollet and Tsybakov [2011];
- 2) the complexity parameter that we will be using in the fixed design setup have been computed under very general design matrix $\mathbb{X}$ in Bellec [2017] but they were only proved to be optimal under the RIP assumption.

Our main result covers more general design matrix than the one satisfying RIP but it turns out that those results do not allow to conclude on the minimax optimality of the associated regularization function beyond the RIP case; moreover and to our knowledge no sharp closed form are available for the computation of this regularization function beyond the RIP case.

The multiplier process and its associated fixed point

Our analysis is based upon the study of the same regularized excess empirical risk quantity as in the random design section: for all $t \in \mathbb{R}^d$,

$$P_N \mathcal{L}_t^\Psi := \left(\|Y - \mathbb{X}t\|_{L_N^2}^2 + \Psi(t) \right) - \left(\|Y - \mathbb{X}t^*\|_{L_N^2}^2 + \Psi(t^*) \right).$$

We use the same quadratic / multiplier decomposition as in the random design case: for all $t \in \mathbb{R}^d$, $P_N \mathcal{L}_t^\Psi = P_N \mathcal{Q}_{t-t^*} + P_N \mathcal{M}_{t-t^*} + \mathcal{R}_{t-t^*}$ where

- $P_N \mathcal{Q}_{t-t^*} := \|\mathbb{X}(t-t^*)\|_{L_N^2}^2$ is the quadratic part
- $P_N \mathcal{M}_{t-t^*} := 2\langle \xi, \mathbb{X}(t^*-t) \rangle$ is the multiplier part
- $\mathcal{R}_{t,t^*} := \Psi(t) - \Psi(t^*)$ is the regularization part.

Contrary to the random design case, in the fixed design setup the only source of randomness is the Gaussian noise ξ, in particular the quadratic process is fully deterministic. Therefore, there is no need to define a fixed point similar to r_Q for a control on the quadratic process. The only fixed point we introduce is a version of the previous multiplier fixed point $r_M(\cdot)$ adapted to the fixed design setup: for $\eta' = 1/8$, let

$$r_{\mathbb{X}}(\rho) := \inf \left(r > 0 : \sigma \ell^* \left(\frac{\rho}{\sqrt{N}} \mathbb{X} B_1^d \cap r B_2^N \right) \leqslant \eta' r^2 \sqrt{N} \right) \tag{3.4.2}$$

where $\mathbb{X}B_1^d := \{\mathbb{X}t : t \in B_1^d\}$ and B_2^N is the unit ball in ℓ_2^N.

Define r_0' as the non-zero solution to the equation $\sigma\ell^*(rB_2^N \cap \mathrm{Im}(\mathbb{X})) = \eta' r^2\sqrt{N}$, where $\mathrm{Im}(\mathbb{X})$ is the image of $\mathbb{X}$ in $\mathbb{R}^d$, i.e.

$$r_0' = \sigma\ell^*(B_2^N \cap \mathrm{Im}(\mathbb{X}))/(\eta'\sqrt{N}) = \sigma\ell^*(B_2^{\mathrm{Rank}(\mathbb{X})})/(\eta'\sqrt{N}) = (\sigma/\eta')\sqrt{\mathrm{Rank}(\mathbb{X})/N}$$

Let ρ_0' be the smallest ρ such that $(\rho/\sqrt{N})\mathbb{X}B_1^d$ contains $r_0'B_2^N \cap \mathrm{Im}(\mathbb{X})$. An argument similar to the one used in the random design case shows that ρ_0' is the smallest radius such that for all $\rho \geqslant \rho_0'$, $r_{\mathbb{X}}(\rho) = r_{\mathbb{X}}(\rho_0') = r_0'$. Finally, we consider $K_0' = \min\{k \in \mathbb{N} : 2^k\rho^* \geqslant 2\rho_0'\}$.

The fixed point function $r_{\mathbb{X}}(\cdot)$ depends on the Gaussian mean width of $\mathbb{X}B_1^d$ intersected with rB_2^N for various radii r. This quantity has been recently controlled in Proposition 2 in Bellec [2017].

Proposition 3.4.3 (Proposition 2 in Bellec [2017])**.** *Let $\mathbb{X} \in \mathbb{R}^{N\times d}$. Assume that the column vectors of $\mathbb{X}$ are in B_2^N. Then, for all $r \geqslant 0$,*

$$\ell^*(\mathbb{X}B_1^d \cap rB_2^N) \leqslant \min\left(4\sqrt{\log(8ed)}, 4\sqrt{\log(8edr^2)}, r\sqrt{\mathrm{Rank}(\mathbb{X})}\right).$$

It follows from some calculations (similar to the one used to obtain the closed form of $r_M(\cdot)$ in (3.1.9)) that there exists an absolute constant $C_{\mathbb{X}}$ such that for all ρ, $r_{\mathbb{X}}(\rho) \leqslant \overline{r_{\mathbb{X}}}(\rho)$ with

$$\overline{r_{\mathbb{X}}}^2(\rho) = C\begin{cases} \frac{\sigma^2\mathrm{Rank}(\mathbb{X})}{N} & \text{if } \rho \geqslant \rho_0' \\ \rho\sigma\sqrt{\frac{1}{N}\log\left(\frac{e\sigma d}{\rho\sqrt{N}}\right)} & \text{if } \sigma^2\log d \leqslant \rho^2 N \leqslant \rho_0'^2 N \\ \rho\sigma\sqrt{\frac{\log(ed)}{N}} & \text{if } \rho^2 N \leqslant \sigma^2\log d. \end{cases} \tag{3.4.3}$$

We see that $\overline{r_{\mathbb{X}}}(\rho)$ in (3.4.3) and $r_M(\rho)$ in (3.1.9) are very close. The only difference comes from the rank of $\mathbb{X}$ and when $N \geqslant \zeta' d$ and $\mathrm{Rank}(\mathbb{X}) \sim d$, the two fixed points r_M and $\overline{r_{\mathbb{X}}}$ are equal up to absolute constants. Furthermore, one can check that there are two absolute constants $0 < C_1 < C_2$ and $C_{\mathbb{X}} = C_{\mathbb{X}}(\rho, d, \sigma, N)$ such that $C_1 \leqslant C_{\mathbb{X}} \leqslant C_2$ and

$$\overline{r_{\mathbb{X}}}^2(\rho) = C_{\mathbb{X}}\min\left(\frac{\sigma^2\mathrm{Rank}(\mathbb{X})}{N}, \rho\sigma\sqrt{\frac{1}{N}\log\left(\frac{e\sigma d}{\rho\sqrt{N}}\right)}, \rho\sigma\sqrt{\frac{\log(ed)}{N}}\right). \tag{3.4.4}$$

We will use $\rho \to \overline{r_{\mathbb{X}}}^2(\rho)$ as a regularization function (up to an absolute constant).

Main result

In this section, we obtain bounds on the rates provided by the regularization function $\Psi(\rho) = c_0'\overline{r_{\mathbb{X}}}^2(\rho)$ for some well-chosen absolute constant c_0' when the design matrix $\mathbb{X}$ satisfies RIP. We first present the minimax rate over ℓ_1^d-balls in the fixed design setup. Such a result was obtained in Rigollet and Tsybakov [2011]. Let us now recall this result in our context (see (5.25) in Section 5.2.2 in Rigollet and Tsybakov [2011]).

Proposition 3.4.4 (Rigollet and Tsybakov [2011]). *Let $\mathbb{X} \in \mathbb{R}^{N\times d}$ be a matrix satisfying RIP(2s). For all $\rho \geqslant 0$, the minimax rate of convergence over ρB_1^d in the Gaussian linear model with fixed design $\mathbb{X}$ in expectation is given by*

$$\min\left(\frac{\sigma^2 \mathrm{Rank}(\mathbb{X})}{N}, \rho\sigma\sqrt{\frac{1}{N}\log\left(\frac{e\sigma d}{\rho\sqrt{N}}\right)}, \rho^2\right). \tag{3.4.5}$$

The latter result holds in expectation whereas we are interested in deviation results. Even though the minimax rate of convergence in deviation over ρB_1^d in the Gaussian linear model under RIP has not been established, we believe that this rate of convergence in deviation is identical to the one given in Proposition 3.4.4. Note that a proof of this fact follows from Section 3 in Lecué and Mendelson [2013] for the minimax lower bound in deviation and from the quadratic / multiplier decomposition of the excess loss together with Lemma 3.4.6 below to show that the ERM over ρB_1^N achieves the minimax bound in deviation. We do not provide the proof here but we will use (3.4.5) as a benchmark for our regularization function.

Theorem 3.4.5. *Let $\mathbb{X} \in \mathbb{R}^{N\times d}$ be such that the column vectors of $\mathbb{X}$ are in B_2^N. Consider the following regularization function:*

$$\rho \geqslant 0 \to \Psi(\rho) = c_0' \overline{r_{\mathbb{X}}}^2(\rho)$$

where $\overline{r_{\mathbb{X}}}(\rho)$ is defined in (3.4.4) and $c_0' \geqslant 2$ is an absolute constant. Then there exist absolute constants κ_1', κ_2' and κ_3' such that for any $t^* \in \mathbb{R}^d$ the RERM $\widehat{t}$ constructed from the data $Y = \mathbb{X}t^* + \xi$:

$$\widehat{t} \in \operatorname*{argmin}_{t\in\mathbb{R}^d}\left(\|Y - \mathbb{X}t\|_{L_N^2}^2 + \Psi(\|t\|_1)\right)$$

satisfies with probability greater than $1 - \kappa_1' \exp\left(-\kappa_2' N \overline{r_{\mathbb{X}}}(\|t^*\|_1)^2/\sigma^2\right)$,

$$\|\mathbb{X}(\widehat{t} - t^*)\|_{L_2^N}^2 \leqslant \kappa_3' \min\left(\frac{\sigma^2\mathrm{Rank}(\mathbb{X})}{N}, \frac{\|t^*\|_1\sigma}{\sqrt{N}}\sqrt{\log\left(\frac{ed\sigma}{\|t^*\|_1\sqrt{N}}\right)}, \|t^*\|_1\sigma\sqrt{\frac{\log(ed)}{N}}, \right).$$

Note that the probability estimate $1 - \kappa_1' \exp\left(-\kappa_2' N \overline{r_{\mathbb{X}}}(\|t^*\|_1)^2/\sigma^2\right) \geqslant 3/4$ only when $\|t^*\|_1 \geqslant \Delta_0'\sigma\sqrt{\log(ed)/N}$ for some absolute constant Δ_0' large enough. Therefore, if (3.4.5) is indeed the minimax rate of convergence over ρB_1^d for the deviation $1 - \delta_N = 3/4$ under RIP then Theorem 3.4.5 proves that $\rho \geqslant 0 \to \Psi(\rho) = c_0' \overline{r_{\mathbb{X}}}^2(\rho)$ is a minimax regularization function for the ℓ_1^d-norm over $\mathbb{R}^d \backslash \Delta_0'\sigma\sqrt{\log(ed)/N} B_1^d$ for the constant confidence regime.

3.4.1. Proof of Theorem 3.4.5

The proof is split into a probabilist part used to identify a high probability event Ω_0' on which the multiplier process is well controlled on the entire space $\mathbb{R}^d$ and a deterministic part where it is proved that, on the event Ω_0', $P_N\mathcal{L}_t^\Psi > 0$ if $\|\mathbb{X}(t - t^*)\|_{L_N^2}^2 \gtrsim \overline{r_{\mathbb{X}}}(\|t^*\|_1)^2$.

Probabilistic control of the multiplier process

The following lemma shows how the fixed point $r_{\mathbb{X}}$ allows to control the multiplier process.

Lemma 3.4.6. *Let $\rho > 0$ and take $\eta' = 1/8$. Then, for all $r \geqslant r_{\mathbb{X}}(\rho)$, with probability greater than $1 - \exp\left(-Nr^2/(128\sigma^2)\right)$, for all $t \in t^* + \rho B_1^d$,*

$$|P_N \mathcal{M}_{t-t^*}| \leqslant \frac{1}{2} \max\left(r^2, \|\mathbb{X}(t-t^*)\|_{L_N^2}^2\right).$$

Proof. Let $r \geqslant r_{\mathbb{X}}(\rho)$. We denote by $B_{\mathbb{X}}^2$ the unit ball associated with the pseudo-metric $\|\mathbb{X}\cdot\|_{L_N^2}$ and $rB_{\mathbb{X}}^2 = \left\{t \in \mathbb{R}^d : \|\mathbb{X}t\|_{L_N^2} \leqslant r\right\}$ its unit ball of radius r. We have

$$\sup_{t \in t^* + \rho B_1^d \cap rB_{\mathbb{X}}^2} |P_N \mathcal{M}_{t-t^*}| = 2 \sup_{t \in t^* + \rho B_1^d \cap rB_{\mathbb{X}}^2} \left|\left\langle \mathbb{X}(t-t^*), \xi \right\rangle_{L_N^2}\right| = \frac{2}{\sqrt{N}} \sup_{v \in V} \langle v, \xi \rangle$$

where $V = [(\rho/\sqrt{N})\mathbb{X}B_1^d] \cap rB_2^N$. Then, it follows from Borell's concentration inequality (cf. Ledoux [2001]) that for all $x > 0$, with probability at least $1 - \exp(-x^2/2)$,

$$\sup_{v \in V} \langle v, \xi \rangle \leqslant \mathbb{E} \sup_{v \in V} \langle v, \xi \rangle + x\sigma(V)$$

where $\sigma(V) := \sup_{v \in V} \sqrt{\mathbb{E}\langle v, \xi \rangle^2} = \sup_{v \in V} \sigma \|v\|_2 \leqslant \sigma r$. Moreover, given that $\xi \sim \mathcal{N}(0, \sigma^2 I_N)$, we have

$$\mathbb{E} \sup_{v \in V} \langle v, \xi \rangle = \sigma \ell^*(V) = \sigma \ell^* \left(\frac{\rho}{\sqrt{N}} \mathbb{X}B_1^d \cap rB_2^N \right) \leqslant \eta' r^2 \sqrt{N}$$

where the last inequality follows from the definition of $r_{\mathbb{X}}(\rho)$ and because $r \geqslant r_{\mathbb{X}}(\rho)$. Gathering all the pieces together, it follows for $x = \eta' r\sqrt{N}/\sigma$, that, with probability at least $1 - \exp(-r^2 N/(128\sigma^2))$,

$$\sup_{t \in t^* + \rho B_1^d \cap rB_{\mathbb{X}}^2} |P_N \mathcal{M}_{t-t^*}| \leqslant 4\eta' r^2 = \frac{1}{2} r^2. \tag{3.4.6}$$

Now, the proof follows from an homogeneity argument. Indeed, let us assume that (3.4.6) holds. Let $t \in t^* + \rho B_1^d$ be such that $\|\mathbb{X}(t-t^*)\|_{L_N^2} > r$ and define $t' := t^* + \alpha_t(t-t^*)$ where $\alpha_t = r/\|\mathbb{X}(t-t^*)\|_{L_N^2}$. Note that $\alpha_t < 1$ and $t' \in t^* + \rho B_1^d \cap rB_{\mathbb{X}}^2$. Hence, it follows from (3.4.6) that $|P_N \mathcal{M}_{t'-t^*}| \leqslant r^2/2$ and so $|P_N \mathcal{M}_{t-t^*}| = |P_N \mathcal{M}_{t'-t^*}|/\alpha_t \leqslant r^2/(2\alpha_t) \leqslant \|\mathbb{X}(\widehat{t} - t^*)\|_{L_N^2}^2/2$. ∎

Deterministic part of the proof

We start with two lemmas on the growth behavior of $\overline{r_{\mathbb{X}}}(\cdot)$. Their proofs are almost identical to the one of Lemmas 3.2.3 and 3.2.4 and are therefore omitted.

Lemma 3.4.7. *Let $\phi' = 4$. If $\phi'\rho \leqslant \rho_0' \min(1, \eta')$, then for any $\rho' \geqslant \phi'\rho$, $\overline{r_{\mathbb{X}}}^2(\rho') > 2\overline{r_{\mathbb{X}}}^2(\rho)$.*

Lemma 3.4.8. *Let $\nu > 0$. If $\nu \geqslant 1$ then $\overline{r_{\mathbb{X}}}(\nu\rho) \leqslant \sqrt{\nu}\overline{r_{\mathbb{X}}}(\rho)$. If $\nu \leqslant 1$ then $\overline{r_{\mathbb{X}}}(\nu\rho) \geqslant \sqrt{\nu}\overline{r_{\mathbb{X}}}(\rho)$.*

To prove Theorem 3.4.5, we use the same argument as in the proof of Theorem 3.1.4 for the random design. Let $\rho^* = 10\|t^*\|_1/\eta'$ and split $\mathbb{R}^d$ into three zones:

- the "central zone" $t^* + \rho^* B_1^d$,
- the intermediate "peeling zone": $\{t \in \mathbb{R}^d : \rho^* < \|t - t^*\|_1 \leqslant 2^{K_0'}\rho^*\}$ – to be considered only when $K_0' \geqslant 1$. This part of $\mathbb{R}^d$ is itself partitioned into K_0' shelves: for $k = 1, \dots, K_0'$, $\{t \in \mathbb{R}^d : 2^{k-1}\rho^* < \|t - t^*\|_1 \leqslant 2^k\rho^*\}$,
- the "exterior zone": $\{t \in \mathbb{R}^d : \|t - t^*\|_1 > 2^{K_0'}\rho^*\}$ on which $r_{\mathbb{X}}$ is constant equal to r_0'.

For all $k = 0, \dots, K_0'$, we denote by A_k' the event on which for all $t \in t^* + 2^k\rho^* B_1^d$,

$$|P_N \mathcal{M}_{t-t^*}| \leqslant \frac{1}{2} \max\left(\overline{r_{\mathbb{X}}}(2^k\rho^*)^2, \|\mathbb{X}(t - t^*)\|_{L_N^2}^2\right).$$

We consider the event $\Omega_0' = A_0' \cap \dots \cap A_{K_0'}'$. It follows from Lemma 3.4.6 and an argument similar to the one in Lemma 3.3.2 that for some absolute constants κ_1', κ_2' and κ_4',

$$\mathbb{P}\left[\Omega_0'\right] \geqslant 1 - \kappa_1' \exp\left(-\kappa_2' N \overline{r_{\mathbb{X}}}(\|t^*\|_1)^2/\sigma^2\right)$$

as long as $\|t^*\|_1 \geqslant \kappa_4'\sigma/\sqrt{N}$ (which is the case when $\|t^*\|_1 \geqslant \Delta_0'\sigma\sqrt{\log(ed)/N}$ for $\Delta_0' \geqslant \kappa_4'$).

Let us now assume for the remaining of the proof that Ω_0' holds. Note that unlike in the random design case, there is no event such as Ω^* in Ω_0' on which the quadratic process is controlled, because, in the deterministic design case this process is deterministic.

Our strategy is to show that $\widehat{t}$ belongs to the "central zone". To that end it is enough to prove that $P_N\mathcal{L}_t^\Psi > 0$ for every $t \in \mathbb{R}^d$ such that $\|t - t^*\|_1 > \rho^*$ because by definition $P_N\mathcal{L}_{\widehat{t}}^\Psi \leqslant 0$.

Let t be in the intermediate peeling zone (which can happen only if $\rho^* \leqslant 2\rho_0'$), say in the k-th shell for some $k \in \{0, \dots, K_0'\}$: $2^{k-1}\rho^* < \|t - t^*\|_1 \leqslant 2^k\rho^*$. In particular $\|t\|_1 > \|t^*\|_1$ and $\mathcal{R}_{t,t^*} > 0$. Therefore, if $\|\mathbb{X}(t - t^*)\|_{L_N^2} \geqslant \overline{r_{\mathbb{X}}}(2^k\rho^*)$ then by Lemma 3.4.6, $|P_N\mathcal{M}_{t-t^*}| \leqslant \|\mathbb{X}(t - t^*)\|_{L_N^2}^2 = P_N\mathcal{Q}_{t-t^*}$ and so $P_N\mathcal{L}_t^\Psi > 0$. Now, if $\|\mathbb{X}(t - t^*)\|_{L_N^2} \leqslant \overline{r_{\mathbb{X}}}(2^k\rho^*)$ then by Lemmas 3.4.6 and 3.4.8, for $c_0' \geqslant 2$,

$$|P_N\mathcal{M}_{t-t^*}| \leqslant \frac{1}{2}\overline{r_{\mathbb{X}}}(2^k\rho^*)^2 < \frac{c_0'}{2}\overline{r_{\mathbb{X}}}^2(\|t - t^*\|_1)$$

and since $\|t\|_1 \geqslant \|t - t^*\|_1 - \|t^*\|_1 \geqslant 4\|t^*\|_1$, and $4\|t^*\|_1 \leqslant \eta'\rho^*/2 \leqslant \eta'\rho_0'$, by Lemma 3.4.7 , one has $c_0'\overline{r_{\mathbb{X}}}^2(\|t\|_1) \geqslant 2c_0'\overline{r_{\mathbb{X}}}^2(\|t^*\|_1)$. As a consequence, $\mathcal{R}_{t,t^*} \geqslant c_0'\overline{r_{\mathbb{X}}}^2(\|t\|_1)/2 > |P_N\mathcal{M}_{t-t^*}|$ and so $P_N\mathcal{L}_t^\Psi > 0$.

Let us now tackle the exterior zone in both cases $\rho^* \leqslant 2\rho_0'$ and $\rho^* > 2\rho_0'$. Let $t \in \mathbb{R}^d$ be such that $\|t - t^*\|_1 > 2^{K_0'}\rho^*$. We have $\mathcal{R}_{t,t^*} \geqslant 0$ because $\Psi(\|t\|_1) = c_0'\overline{r_{\mathbb{X}}}^2(\|t\|_1) = c_0'r_0'^2 \geqslant c_0'\overline{r_{\mathbb{X}}}^2(\|t^*\|_1) = \Psi(\|t^*\|_1)$. Let $t' = t^* + \alpha_t(t - t^*)$ for some $0 < \alpha_t < 1$ be such that $\|t' - t^*\|_1 = 2^{K_0'}\rho^*$. By definition of K_0' and ρ_0', we have $\|\mathbb{X}(t' - t^*)\|_{L_N^2} \geqslant r_0'$. Therefore, since $A_{K_0'} \subset \Omega_0'$, we have $|P_N\mathcal{M}_{t'-t^*}| \leqslant (1/2)\|\mathbb{X}(t' - t^*)\|_{L_N^2}^2$ which implies that $P_N\mathcal{Q}_{t'-t^*} + P_N\mathcal{M}_{t'-t^*} > 0$ and therefore by an homogeneity argument that $P_N\mathcal{Q}_{t-t^*} + P_N\mathcal{M}_{t-t^*} > 0$. Finally, given that $\mathcal{R}_{\widehat{t},t^*} \geqslant 0$ we conclude that $P_N\mathcal{L}_t^\Psi > 0$.

This proves that $\widehat{t}$ lies in the central zone in both cases $\rho^* \leqslant 2\rho_0'$ and $\rho^* > 2\rho_0'$. But, now given that $A_0' \subset \Omega_0'$, we have

$$|P_N \mathcal{M}_{\widehat{t}-t^*}| \leqslant \frac{1}{2} \max\left(\overline{r_{\mathbb{X}}}(\rho^*)^2, \|\mathbb{X}(\widehat{t}-t^*)\|_{L_N^2}^2\right).$$

If $\|\mathbb{X}(\widehat{t}-t^*)\|_{L_N^2}^2 \leqslant \overline{r_{\mathbb{X}}}(\rho^*)^2$ the proof is over and otherwise $|P_N \mathcal{M}_{\widehat{t}-t^*}| \leqslant (1/2)\|\mathbb{X}(\widehat{t}-t^*)\|_{L_N^2}^2$ which implies that $\|\widehat{t}-t^*\|_{L_N^2}^2 \leqslant 2\Psi(\|t^*\|_1) \leqslant 2c_0'\overline{r_{\mathbb{X}}}(\rho^*)^2$ because

$$0 \geqslant P_N \mathcal{L}_{\widehat{t}}^{\Psi} \geqslant \frac{1}{2}\|\widehat{t}-t^*\|_{L_N^2}^2 + \Psi(\|\widehat{t}\|_1) - \Psi(\|t^*\|_1).$$

This proves, on the event Ω_0', that $\|\widehat{t}-t^*\|_{L_N^2}^2 \leqslant 2c_0'\overline{r_{\mathbb{X}}}(\rho^*)^2$. ■

Chapter 4

Improvements on an online convex optimization algorithm: MetaGrad

This chapter comes from joint work with Tim van Erven and Dirk van der Hoeven.

In this chapter, we tackle the Online Convex Optimization framework: the losses can be any convex function, and the goal is to minimize the (cumulative) regret. The methods to use, and the optimal performance, depend on the curvature of the losses: additional assumptions on it (such as strong convexity) allow a better performance; however, the algorithms that achieve optimal bounds are often specific and need to be tuned with a priori knowledge on the convexity parameters.

We focus on a recent algorithm, MetaGrad, introduced by van Erven and Koolen [2016], that, interestingly, is adaptive to the level of curvature: it provides optimal or nearly-optimal bounds for various types of convexities, without requiring the user to tune it beforehand using a priori knowledge on the curvature.

Our contributions are twofold. First, we present modifications of MetaGrad that reduce computation time, and exhibit some corresponding theoretical bounds. Secondly, we tackle another framework, the batch setting, where i.i.d. losses are suffered, and we modify the classical "online-to-batch" adaptation of MetaGrad (and of another algorithm, Online Newton Step) to improve the bounds and take benefits of the adaptivity properties of MetaGrad, in the case of strongly convex losses.

4.1. The MetaGrad algorithm

4.1.1. General presentation

Online Convex Optimization. In this chapter, we will go beyond linear regression with convex losses to tackle a more general framework: online convex optimization. In this setup, we still provide round after round a vector w_t in a domain $\mathcal{K} \in \mathbb{R}^d$ and then get a loss $f_t(w_t)$, that we want to minimize, but the only specification about this loss is that is is convex.

In the particular setup of linear regression using the square loss, $f_t(w_t) = (w_t^\top x_t - y_t)^2$, where y_t is the observation, and x_t is the vector of explanatory variables.

The domain of interest $\mathcal{K} \subset \mathbb{R}^d$ is convex; we will assume it is closed. It is generally fixed, but in some cases it can vary over time (cf. Section 4.2.2 for an example).

We will assume in this chapter that the number of rounds T is known beforehand.

We will first focus on the case where the f_t are deterministic and unknown in advance: they can be any convex functions. The goal is then to minimize the following regret:

$$\sum_{t=1}^{T} f_t(w_t) - \min_{u \in \mathcal{K}} \sum_{t=1}^{T} f_t(u).$$

Another framework in Section 4.3. We will see in Section 4.3 a batch stochastic framework, where the f_t are random functions, i.i.d., drawn from an unknown distribution $\mathbb{P}$. Then, the goal will be (leaning on a learning sample known in advance) to build an algorithm which outputs a vector $\widehat{w}_T$ with a small expected regret, i.e., which tries to minimize:

$$E_{f, f_1, \ldots, f_T \sim \mathbb{P}}[f(\widehat{w}_T)] - \min_{u \in \mathcal{K}} E_{f \sim \mathbb{P}}[f(u)]$$

where the first expectation is defined with respect to $f, f_1, \ldots, f_T$.

First-order information. We recall that the mere hypothesis of convexity (without any smoothness assumption) guarantees the existence of a non-empty subgradient at any point in the interior $\mathring{\mathcal{K}}$ of $\mathcal{K}$:

$$\forall\, x \in \mathring{\mathcal{K}}, \quad \partial f_t(x) \neq \emptyset, \text{ where } \partial f_t(x) := \big\{v : \ \forall\, y \in \mathcal{K}, \ f_t(y) \geqslant f_t(x) + v^\top (y - x)\big\}.$$

We will focus on the case of first-order information, where one has access after time t to an element of the subgradient of the loss: $g_t \in \partial f_t(w_t)$ (we will assume that this subgradient will be non-empty even if $w_t \notin \mathring{\mathcal{K}}$, which is the case for instance if the f_t are differentiable).

The process of online convex optimization with first-order information is summarized in Setting 6.

Setting 6 Online convex optimization (with first-order information) framework

for $t = 1, 2, \ldots, T$:

1. Play $w_t \in \mathcal{K}_t$
2. The environment picks a convex loss function $f_t : \mathcal{K}_t \to \mathbb{R}$
3. Incur cumulated loss by $f_t(w_t)$ and observe (sub)gradient $g_t \in \partial f_t(w_t)$

end for

Impact of the loss functions. Let us recall some elements about the losses (see Section 2.3.2 of Chapter 2 for more details). The loss functions, and in particular their curvature, impact strongly the results one can get, and the efficiency of algorithms. For general convex losses, the "Online Gradient Descent" algorithm, introduced by Zinkevich [2003] and based on the following update: $w_{t+1} = w_t - \eta_t \nabla f_t(w_t)$, achieves the optimal regret $O(\sqrt{T})$ when the parameter η_t is of order $1/\sqrt{t}$. In the case of strongly convex losses, the same Online Gradient Descent algorithm, with the parameter η_t chosen of order $1/t$, achieves a regret of order $O(\log T)$. As for the intermediate case of exp-concave losses, the "Online Newton Step" algorithm, introduced by Hazan et al. [2007], guarantees a bound $O(d \log T)$ on the regret. As this algorithm will play a key role in this chapter, we detail in Algorithm 7 a version of Online Newton Step adapted to the context of our works.

The question of adaptivity, with respect to the degree of curvature (type of convexity, but also for instance strong convexity parameter or exp-concavity parameter) is therefore very important, as much for theoretical problems as for practical applications (for which this degree of curvature is often unknown); that is precisely one of the key features of the algorithm we focus on in this chapter: the MetaGrad algorithm.

The MetaGrad algorithm. The MetaGrad algorithm is an online convex optimization algorithm introduced in van Erven and Koolen [2016]. It aims at adaptivity on two aspects: the type of curvature (discussed in the previous paragraph) and the optimal learning rate. To do so, it relies on two algorithmic layers. The first layer is composed of several "parallel" algorithms, nicknamed the "slaves", that only differ by their learning rate, and are close in spirit to Online Newton Step. The second layer is composed of a meta-algorithm, nicknamed

Algorithm 7 Online Newton Step

Input: Learning rate η. Parameters ε (initialization) and m (update). Starting point w_1. Domain $\mathcal{K}$.

Initialization:
Get the starting point w_1 and the initial covariance matrix inverse $A_0 = \varepsilon I_d$.

for $t = 1, 2, \ldots$

1. Observe (sub)gradient $g_t \in \partial f_t(w_t)$
2. $A_t = A_{t-1} + m g_t g_t^\top$
3. $\widetilde{w}_{t+1} = w_t - (A_t)^{-1}\big(\eta g_t\big)$
4. $w_{t+1} = \underset{u \in \mathcal{K}}{\operatorname{argmin}} \ (u - \widetilde{w}_{t+1}) A_t (u - \widetilde{w}_{t+1})$

the "master", that will perform an aggregation (using modified exponential weights) upon the slaves' forecasts, and thus will be able, in some sense, to "learn the learning rate".

MetaGrad is detailed in Algorithm 8 and Algorithm 9. It uses in the master algorithm a surrogate loss, which depends on the output of the algorithm w_t:

$$\ell_t^\eta(u) = \eta(u - w_t)^\top g_t + \Big(\eta(u - w_t)^\top g_t\Big)^2. \tag{4.1.1}$$

Several remarks can be made on the algorithms.

First, the slave update is very close to the Online Newton Step update, but here the gap of the gradient measurement point w_t and the slave output w_t^η leads to the addition of an extra term $2\eta^2 (A_t^\eta)^{-1} g_t g_t^\top (w_t^\eta - w_t)$ to try to counterbalance this gap.

Also notice that the projection in Step 5 of Algorithm 9 is not the usual Euclidian projection, but uses instead a Mahalanobis norm that depends on the data and on the previous choices of w_t, and is a metric more adapted to the situation (in particular it will be useful in the analysis). The matrix corresponding to the dot product of this norm, A_t^η, is denoted by $\big(\Sigma_{t+1}^\eta\big)^{-1}$ in the original paper van Erven and Koolen [2016].

As for the master algorithm, its update uses "tilted" exponential weights (in the sense that they are multiplied by their learning rate), giving larger weights to the largest η; this will be motivated by the analysis (but one could also add that in practice, favouring a bit the large learning rate is often a good idea, the smallest learning rate being generally too conservative, at least for well-behaved data).

4.1.2. Analysis

We recall in this Section a major theorem of van Erven and Koolen [2016], and afterwards some analyses, and intermediate results leading to it, upon which we will build the arguments of the next sections.

Algorithm 8 MetaGrad Master

Input: Number of time steps T. Diameter of the domain D: $D = \sup_{u,v\in\mathcal{K}} \|v-u\|_2$.
Uniform bound G on the (sub)gradients: $G \geqslant \sup_t \sup_{u\in\mathcal{K}} \max_{g_t\in\partial f_t(u)} \|g_t\|_2$.
Starting point w_1. Initial covariance matrix inverse $A_0 = \varepsilon I_d$.

Initialization:
1. Define the grid of learning rates: $\eta_i = 2^{-i}/(5DG)$ for $i = 0, 1, \ldots, \lceil \log_2(T)/2 \rceil$ with prior weights $\pi_1^{\eta_i} = \frac{1+1/(1+\lceil \log_2(T)/2 \rceil)}{(i+1)(i+2)}$.
2. Launch the corresponding MetaGrad Slaves algorithms (Algorithm 9).
3. Send the starting point w_1 and the initial covariance matrix inverse $A_0 = \varepsilon I_d$ to the slaves.

for $t = 1, 2, \ldots$:
1. Get prediction $w_t^\eta \in \mathcal{K}$ of slave η (Algorithm 9) for each η.
2. Play $w_t = \frac{\sum_\eta \pi_t^\eta \eta w_t^\eta}{\sum_\eta \pi_t^\eta \eta} \in \mathcal{K}$.
3. Observe gradient $g_t \in \partial f_t(w_t)$.
4. Update $\pi_{t+1}^\eta = \frac{\pi_t^\eta e^{-\ell_t^\eta(w_t^\eta)}}{\sum_\nu \pi_t^\nu e^{-\ell_t^\nu(w_t^\nu)}}$ for all η *where the surrogate loss* ℓ_t^η *is defined in* (4.1.1)

Algorithm 9 MetaGrad Slave

Input: Learning rate $0 < \eta \leqslant \frac{1}{5DG}$. Domain $\mathcal{K}$.

Initialization:
Get the starting point $w_1^\eta := w_1$ and the initial covariance matrix inverse $A_0^\eta := A_0 = \varepsilon I_d$ from MetaGrad Master (Algorithm 8).

for $t = 1, 2, \ldots$
1. Send w_t^η to MetaGrad Master (Algorithm 8)
2. Observe (sub)gradient $g_t \in \partial f_t(w_t)$ *(at* ***master*** *point* w_t*)*

Update:
3. $A_t^\eta = \varepsilon I_d + 2\eta^2 \sum_{s=1}^t g_s g_s^\top$
4. $\widetilde{w}_{t+1}^\eta = w_t^\eta - (A_t^\eta)^{-1}\big(\eta g_t + 2\eta^2 g_t g_t^\top (w_t^\eta - w_t)\big)$
5. $w_{t+1}^\eta = \operatorname*{argmin}_{u\in\mathcal{K}} (u - \widetilde{w}_{t+1}^\eta) A_t^\eta (u - \widetilde{w}_{t+1}^\eta)$

A bound on the MetaGrad algorithm .

Let us recall Theorem 1 of van Erven and Koolen [2016].

Theorem 4.1. *Let $g_t = \nabla f_t(w_t)$ and, for any u in $\mathcal{K}$, $V_T^u = \sum_{t=1}^T \left((u - w_t)^\top g_t\right)^2$. Then the regret of MetaGrad is simultaneously bounded by $O\left(\sqrt{T \log\log(T)}\right)$ and by*

$$\sum_{t=1}^T f_t(w_t) - \sum_{t=1}^T f_t(u) \leqslant \sum_{t=1}^T (w_t - u)^\top g_t = O\left(\sqrt{V_T^u d \log(T)} + d\log(T)\right)$$

The inequality in this theorem is a direct consequence of the convexity and of the subgradient definition. We recall in the remaining of this section some of the steps to upper bound for any u the quantity $\tilde{R}_t^u := \sum_{t=1}^T g_t^\top (w_t - u)$.

A useful lemma for the master algorithm analysis.

Lemma 4.2. *Define $\Phi_t := \sum_\eta \pi_1^\eta e^{-\sum_{s=1}^t \ell_s^\eta(w_s^\eta)}$.*
When $\eta \in [0, \frac{2}{3DG}]$, the master algorithm guarantees $1 = \Phi_0 \geqslant \Phi_1 \geqslant \ldots \geqslant \Phi_T$.

Proof. Recall that $\ell_s^\eta(u) = \eta(u - w_s)^\top g_s + \left(\eta(u - w_s)^\top g_s\right)^2$. The Cauchy-Schwarz inequality gives: $|(u - w_s)^\top g_s| \leqslant \|u - w_s\|_2 \|g_s\|_2 \leqslant DG$.
Applying the inequality $e^{x - x^2} \leqslant 1 + x$, true for any $x \geqslant -2/3$, we have that

$$e^{-\ell_s^\eta(w_s^\eta)} \leqslant 1 + \eta(w_s - w_s^\eta)^\top g_s \quad \text{for any } \eta \in [0, \frac{2}{3DG}].$$

This shows that the potential is non-increasing:

$$\begin{aligned}\Phi_{t+1} - \Phi_t &= \sum_\eta \pi_1^\eta e^{-\sum_{s=1}^t \ell_s^\eta(w_s^\eta)} \left(e^{-\ell_{t+1}^\eta(w_{t+1}^\eta)} - 1\right) \\ &\leqslant \sum_\eta \pi_1^\eta e^{-\sum_{s=1}^t \ell_s^\eta(w_s^\eta)} \eta(w_{t+1} - w_{t+1}^\eta)^\top g_{t+1} = 0,\end{aligned}$$

where the final equality is due to the definition of w_{t+1}.

The initialization $\Phi_0 = 1$ is immediate since $\sum_\eta \pi_1^\eta = 1$. ∎

Slaves algorithms. Now, following Hazan et al. [2007] (see Lemma 4.8 below and its proof) and van Erven and Koolen [2016], let us analyse the slaves performance.

For readability convenience, in all the following, we will write A_t instead of A_t^η when there is no ambiguity.

Lemma 4.3. *For $\eta \in (0, \frac{1}{5DG}]$ we have that the regret of the slave algorithm is bounded by:*

$$\sum_{t=1}^T \ell_t^\eta(w_t^\eta) \leqslant \sum_{t=1}^T \ell_t^\eta(u) + \frac{\varepsilon}{2}\|u - w_1^\eta\|_2^2 + \frac{1}{2}\log\left(\det\left(I_d + \frac{2\eta^2}{\varepsilon}\sum_{t=1}^T g_t g_t^\top\right)\right) \tag{4.1.2}$$

for any $u \in \mathcal{K}$.

Proof. In the first part of the proof, we will obtain a useful form for $\sum_{t=1}^T \ell_t^\eta(w_t^\eta) - \sum_{t=1}^T \ell_t^\eta(u)$, *using the updates of the algorithm.*

We denote $M_t = g_t g_t^\top$.

$$\begin{aligned}
\ell_t^\eta(w_t^\eta) - \ell_t^\eta(u) &= -\eta(w_t - w_t^\eta)^\top g_t + \eta^2 (w_t - w_t^\eta)^\top M_t (w_t - w_t^\eta) + \\
&\qquad \eta(w_t - u)^\top g_t - \eta^2 (w_t - u)^\top M_t (w_t - u) \\
&= \eta(w_t^\eta - u)^\top g_t + \eta^2 {w_t^\eta}^\top M_t w_t^\eta - 2\eta^2 {w_t^\eta}^\top M_t w_t - \eta^2 u^\top M_t u + 2\eta^2 u^\top M_t w_t \\
&= \eta(w_t^\eta - u)^\top g_t + \eta^2 \left(2{w_t^\eta}^\top M_t w_t^\eta - 2 w_t^\top M_t w_t^\eta + 2u^\top M_t w_t^\top - 2u^\top M_t w_t^\eta \right) - \\
&\qquad \eta^2 \left(2{w_t^\eta}^\top M_t w_t^\eta + u^\top M_t u - 2u^\top M_t w_t^\eta \right) \\
&= \eta(w_t^\eta - u)^\top g_t + 2\eta^2 (w_t^\eta - u)^\top M_t (w_t^\eta - w_t) - \eta^2 (u - w_t^\eta)^\top M_t (u - w_t^\eta) \\
&= \eta(w_t^\eta - u)^\top \tilde{g}_t - \eta^2 (u - w_t^\eta)^\top M_t (u - w_t^\eta),
\end{aligned}$$

where $\tilde{g}_t = \left(1 + 2\eta g_t^\top (w_t^\eta - w_t)\right) g_t$.

One has $\tilde{w}_{t+1}^\eta = w_t^\eta - \eta A_t^{-1} \tilde{g}_t$. The following argument is then similar to the Online Newton Step analysis (cf. proof of Lemma 4.8). One can use the following classical property of projections:

$$\begin{aligned}
(u - w_{t+1}^\eta)^\top A_t (u - w_{t+1}^\eta) &\leqslant (u - \tilde{w}_{t+1}^\eta)^\top A_t (u - \tilde{w}_{t+1}^\eta) \\
&= (u - w_t^\eta + \eta A_t^{-1} \tilde{g}_t)^\top A_t (u - w_t^\eta + \eta A_t^{-1} \tilde{g}_t) \\
&= \eta^2 \tilde{g}_t^\top A_t^{-1} \tilde{g}_t + (u - w_t^\eta)^\top A_t (u - w_t^\eta) - 2\eta (w_t^\eta - u)^\top \tilde{g}_t.
\end{aligned}$$

Thus, one gets:

$$\eta(w_t^\eta - u)^\top \tilde{g}_t \leqslant \frac{1}{2} \left(\eta^2 \tilde{g}_t^\top A_t^{-1} \tilde{g}_t + (u - w_t^\eta)^\top A_t (u - w_t^\eta) - (u - w_{t+1}^\eta)^\top A_t (u - w_{t+1}^\eta) \right).$$

Therefore, we have

$$\begin{aligned}
\ell_t^\eta(w_t^\eta) - \ell_t^\eta(u) \leqslant & \frac{\eta^2}{2} \tilde{g}_t^\top A_t^{-1} \tilde{g}_t + \frac{1}{2}(u - w_t^\eta)^\top A_t (u - w_t^\eta) - \frac{1}{2}(u - w_{t+1}^\eta)^\top A_t (u - w_{t+1}^\eta) \\
& - \eta^2 (u - w_t^\eta)^\top M_t (u - w_t^\eta)
\end{aligned} \tag{4.1.3}$$

This bound does not make use of the real values and construction of the A_s *(it is true for any symmetric positive matrix* A_s*): this will be the key element for comparisons in the "Sketching" Section 4.2.2.*

One can divide the right-hand side of the "cumulated" version of (4.1.3) into two parts:

$$\sum_{t=1}^T \ell_t^\eta(w_t^\eta) - \sum_{t=1}^T \ell_t^\eta(u) \leqslant R_G + R_D$$

with $R_G = \sum_{t=1}^T \frac{\eta^2}{2} \tilde{g}_t^\top A_t^{-1} \tilde{g}_t$

and $R_D = \sum_{t=1}^T \left(\frac{1}{2}(u - w_t^\eta)^\top A_t (u - w_t^\eta) - \frac{1}{2}(u - w_{t+1}^\eta)^\top A_t (u - w_{t+1}^\eta) - \eta^2 (u - w_t^\eta)^\top M_t (u - w_t^\eta) \right).$ (4.1.4)

The notations R_G and R_D, taken from Luo et al. [2016], stand respectively for "gradient (part of the) regret" and "diameter (part of the) regret".

In the second part of the proof, we study and bound more precisely R_G and R_D to get the desired result.

Recalling that $A_{t+1} = A_t + 2\eta^2 M_{t+1}$ and $A_0 = \varepsilon I_d$, one gets the following telescopic sum:

$$\begin{aligned} R_D &= \sum_{t=1}^{T} \left(\frac{1}{2}(u - w_t^\eta)^\top A_t (u - w_t^\eta) - \frac{1}{2}(u - w_{t+1}^\eta)^\top A_{t+1}(u - w_{t+1}^\eta) + \right. \\ &\qquad \left. \eta^2 (u - w_{t+1}^\eta)^\top M_{t+1}(u - w_{t+1}^\eta) - \eta^2 (u - w_t^\eta)^\top M_t (u - w_t^\eta) \right) \\ &= \sum_{t=1}^{T} (u - w_1^\eta)^\top (\frac{1}{2} A_1 - \eta^2 M_1)(u - w_1^\eta) - \frac{1}{2}(u - w_{T+1}^\eta)^\top A_T (u - w_{T+1}^\eta) \end{aligned}$$

Therefore:

$$R_D \leqslant \frac{\varepsilon}{2} \|u - w_1^\eta\|_2^2 \tag{4.1.5}$$

Notice that w_{T+1}^η is only an artifact of computation that does not need to be known, since $(u - w_{T+1}^\eta)^\top A_T (u - w_{T+1}^\eta)/2$ is simply lower bounded by 0 in the computations.

For $\eta \in (0, \frac{1}{5GD}]$, the Cauchy-Schwarz inequality yields:

$$1 + 2\eta g_t^\top (w_t^\eta - w_t) \leqslant 1 + 2\eta \|g_t\|_2 \|w_t^\eta - w_t\|_2 \leqslant 1 + 2\eta GD \leqslant \frac{7}{5},$$

and similarly $1 + 2\eta g_t^\top (w_t^\eta - w_t) \geqslant 3/5$. So for all t,

$$\frac{\eta^2}{2} \tilde{g}_t^\top A_t^{-1} \tilde{g}_t \leqslant \left(\frac{7}{5}\right)^2 \frac{\eta^2}{2} g_t^\top A_t^{-1} g_t \leqslant \eta^2 g_t^\top A_t^{-1} g_t.$$

Let us recall Lemma 11 of Hazan et al. [2007].

Lemma 4.4. *Let $u_t \in \mathbb{R}^d$ for $t = 1, \ldots, T$ such that for some $r > 0$, $\|u_t\|_2 \leqslant r$. Define $V_t = \sum_{s=1}^t u_s u_s^\top + \varepsilon I_d$ (and $V_0 = \varepsilon I_d$). Then:*

$$\sum_{t=1}^{T} u_t^\top V_t^{-1} u_t \leqslant \log\left(\frac{\det(V_T)}{\det(V_0)}\right) \leqslant d \log\left(\frac{r^2 T}{\varepsilon} + 1\right).$$

Applying this lemma with $u_t = \sqrt{2}\eta g_t^\top$, one gets that:

$$R_G := \sum_{t=1}^{T} \frac{\eta^2}{2} \tilde{g}_t^\top A_t^{-1} \tilde{g}_t \leqslant \frac{1}{2} \sum_{t=1}^{T} 2\eta^2 g_t^\top A_t^{-1} g_t \leqslant \frac{1}{2} \log\left(\det\left(I_d + \frac{2\eta^2}{\varepsilon} \sum_{t=1}^{T} g_t g_t^\top\right)\right) \tag{4.1.6}$$

The result follows from (4.1.4), (4.1.5) and (4.1.6).

■

Combined analyses. Those results on the master algorithm and on the slaves algorithms allow to bound for any u the quantity $\tilde{R}_t^u := \sum_{t=1}^T g_t^\top (w_t - u)$ and therefore, by convexity, $\sum_{t=1}^T f_t(w_t) - f_t(u)$.

Lemma 4.5. *Starting from an arbitrary point $w_1 \in \mathcal{K}$, apply T iterations of MetaGrad, using as covariance matrix initialization $A_0^\eta = \varepsilon I_d$. Then, for any $\eta \in (0, \frac{1}{5DG}]$, and for any $u \in \mathcal{K}$:*

$$\tilde{R}_T^u \leqslant \eta \sum_{t=1}^T \left((u - w_t)^\top g_t\right)^2 + \frac{1}{\eta}\left(\frac{\varepsilon}{2}\|w_1 - u\|_2^2 - \log(\pi_1^\eta) + \frac{1}{2}\log\left(\det\left(I_d + 2\frac{\eta^2}{\varepsilon}\sum_{t=1}^T g_t g_t^\top\right)\right)\right)$$

Proof. By Lemma 4.2, one has $\log(\Phi_T) \leqslant 0$. Moreover, by Lemma 4.3,

$$\begin{aligned}\log(\Phi_T) &\geqslant \log\left(\pi_1^\eta\right) - \sum_{t=1}^T \ell_t^\eta(w_t^\eta) \\ &\geqslant \log\left(\pi_1^\eta\right) - \sum_{t=1}^T \ell_t^\eta(u) - \frac{\varepsilon}{2}\|u - w_1\|_2^2 - \frac{1}{2}\log\left(\det\left(I_d + \frac{2\eta^2}{\varepsilon}\sum_{t=1}^T g_t g_t^\top\right)\right)\end{aligned}$$

Then, combining the two previous inequalities and using the definition of $\ell_t^\eta(\cdot)$ leads to:

$$\eta\tilde{R}_T^u \leqslant \frac{\varepsilon}{2}\|u - w_1\|_2^2 - \log\left(\pi_1^\eta\right) + \eta^2 \sum_{t=1}^T \left((u - w_t)^\top g_t\right)^2 + \frac{1}{2}\log\left(\det\left(I_d + \frac{2\eta^2}{\varepsilon}\sum_{t=1}^T g_t g_t^\top\right)\right)$$

and dividing by η gives the result. ∎

It remains then to choose correctly the grid of the η's to obtain guarantees on the regret, as it is shown in Theorem 7 of van Erven and Koolen [2016], which leads to Theorem 4.1, and in Section 4.3.3.

4.2. Speeding MetaGrad up

Motivation. One limitation of MetaGrad is computational: it is its speed. In high dimension d, the matrices $d \times d$ are heavy to manage (in particular as far as the projection step is concerned). It is therefore natural to try and modify the original MetaGrad (called "full MetaGrad" in van Erven and Koolen [2016]) to accelerate it, if possible without (or at least without too much) spoiling its theoretical guarantees and practical performance.

A previous attempt. A first attempt is made in van Erven and Koolen [2016]: "Diagonal MetaGrad". It consists in replacing the covariance matrices $\Sigma_{t+1} = A_t^{-1}$ by diagonal matrices, which decreases strongly the computing time. The paper provides an analysis for "Diagonal MetaGrad" quite similar to the full version, though it does not provide all the same theoretical bounds. Some experiments that we made on real datasets (not detailed in this thesis) show some losses of accuracy in the case of "Diagonal MetaGrad" compared to the full version.

Our work. To speed-up MetaGrad, we suggest exploiting the similarity between the MetaGrad slave algorithm and the Online Newton Step algorithm (introduced in Hazan et al. [2007]) to adapt two ideas introduced for Online Newton Step in Luo et al. [2016]. The first one is to use a data-dependent evolving domain, to make the projection easier. The second idea is to use sketching techniques to reduce the dimension of matrices at stake. Thus, in this section, we adapt some of the corresponding proofs and results of Luo et al. [2016] to MetaGrad.

We focus on losses based on linear regression: $f_t : w_t \mapsto \ell_t(w_t^\top x_t)$ with known feature vector x_t, where the $\ell_t : \mathbb{R} \mapsto \mathbb{R}$ are convex and differentiable.

4.2.1. Changing the domain to accelerate the projection

Speed key step: the projection. The bottleneck of MetaGrad computations is often the projection step:

$$w_{t+1}^\eta = \operatorname*{argmin}_{u \in \mathcal{K}} (u - \widetilde{w}_{t+1}^\eta) A_t^\eta (u - \widetilde{w}_{t+1}^\eta).$$

This projection is not bound to occur at each step (if $\tilde{w}_{t+1}^\eta \in \mathcal{K}$, then the projection step is just skipped: $w_{t+1}^\eta = \tilde{w}_{t+1}^\eta$); possibly, if $\mathcal{K}$ is large enough, it may just never occur. But when it is needed, it is the most complex step of the algorithm.

A possible improvement. An acceleration can be obtained if one defines a data-dependent domain that evolves along time. We assume that an upper bound L is known on $\ell_t'(z)$ for all $z \in \mathbb{R}$. The data-dependent domain will depend on a size C that must be chosen beforehand such that $C \leqslant 5DG/3L$. Then, for the t-th forecast, given the feature vector x_t, one works in the domain $\mathcal{K}_t := \{w : \ |w^\top x_t| \leqslant C\}$.

The interest of this domain change is the following quick-to-compute, closed-form expression provided by Luo et al. [2016] to project on $\mathcal{K}_t$:

$$\begin{aligned} w_{t+1}^\eta &= \min_{w \in \mathcal{K}_{t+1}} (w - \tilde{w}_{t+1}^\eta)^\top A_t (w - \tilde{w}_{t+1}^\eta) \\ &= \tilde{w}_{t+1}^\eta - \frac{\tau_C(x_{t+1}^\top \tilde{w}_{t+1}^\eta)}{x_{t+1}^\top A_t^{-1} x_{t+1}} A_t^{-1} x_{t+1} \end{aligned}$$

where $\tau_C(y) = \operatorname{sign}(y) \max\{|y| - C, 0\}$.

As the algorithm gives guarantees at time t compared to vectors of $\mathcal{K}_t$, it is natural then to seek guarantees comparing the algorithm to the vectors that belong to all the $\mathcal{K}_t$. Thus, the comparison $\mathcal{K}$ set will be the intersection of all these domains:

$$\mathcal{K} = \bigcap_{t=1..T} \mathcal{K}_t = \{w : \ \forall\, t = 1..T, \ |w^\top x_t| \leqslant C\}$$

Using the Cauchy-Schwarz inequality, one can see that $\mathcal{K}$ contains *a minima* the Euclidean ball $\{w : \|w\|_2 \leqslant C/(\max_t(\|x_t\|_2))\}$.

One can check that Lemma 4.2 still holds for $\eta \in [0, 1/5DG]$. This comes from the fact that

$$w_t^\top g_t = w_t^\top x_t \, \ell_t'(w_t^\top x_t),$$

which leads for any s to:

$$|\eta(w_s - w_s^\eta)^\top g_s| \leqslant |\eta L(w_s - w_s^\eta)^\top x_s| \leqslant 2\eta LC.$$

Since $C \leqslant 5DG/3L$ and $\eta \leqslant 1/5DG$ one has $\eta(w_s - w_x^\eta)^\top g_s \geqslant -2/3$. Therefore one can still apply $e^{-\ell_s^\eta(w_s^\eta)} \leqslant 1 + \eta(w_s - w_s^\eta)^\top g_s$, which leads to the lemma.

One can easily check that the other parts of the analysis of MetaGrad remain unchanged by this change of domain.

4.2.2. Sketching to deal with smaller matrices

In this section, we consider modifications for the slaves algorithm, the master algorithm (which is fast enough to compute) being unchanged.

Idea of sketching: let us contract time!

The computational complexity of the algorithm increases substantially with the dimension d. The largest matrices to maintain are the $d \times d$ covariance matrices $\Sigma_{t+1}^\eta = A_t^{-1}$. In high dimension, computations involving A_t^{-1} are costly.

To speed the computations up, first, notice that A_t can be written as:

$$A_t = \varepsilon I_d + G_t^\top G_t$$

where the $t \times d$ matrix G_t is such that its $i-$th row is $\sqrt{2\eta} g_i^\top$. One can then write the following, as a special case of the Woodbury formula:

$$A_t^{-1} = \frac{1}{\varepsilon}\left(I_d - G_t^\top \left(\varepsilon I_t + G_t G_t^\top\right)^{-1} G_t\right).$$

This computation involves the inverse of the $t \times t$ square matrix $\varepsilon I_t + G_t G_t^\top$. The idea of sketching is then to approximate $G_t G_t^\top$ by a matrix $S_t S_t^\top$ where S_t is a $m \times d$ matrix, with $m < t$ and, as much as possible, $m \ll d$. We want to contract time, in some sense...

Then, replacing A_t by $\tilde{A}_t = \varepsilon I_d + S_t S_t^\top$, and denoting $H_t = (\varepsilon I_m + S_t S_t^\top)^{-1}$ (which is only an $m \times m$ matrix), one can apply again the Woodbury formula:

$$\tilde{A}_t^{-1} = \frac{1}{\varepsilon}\left(I_d - S_t^\top H_t S_t\right)$$

If one uses the projection seen in Section 4.2.1, and denotes:

$$\gamma_t = \tau_C(x_{t+1}^\top \tilde{w}_{t+1}^\eta)/(x_{t+1}^\top x_{t+1} - x_{t+1}^\top S_t^\top H_t S_t x_{t+1})$$

then (with $\tilde{g}_t = \left(1 + 2\eta g_t^\top (w_t^\eta - w_t)\right) g_t$) the slave updates now become:

$$\begin{aligned}\tilde{w}_{t+1}^\eta &= w_t - \frac{\eta}{\varepsilon}\tilde{g}_t^\top (I_d - S_t^\top H_t S_t)\\ w_{t+1} &= \tilde{w}_{t+1}^\eta - \gamma_t(x_{t+1} - S_t^\top H_t S_t x_{t+1}).\end{aligned}$$

Operations involving $S_t\tilde{g}_t$ or $S_t x_{t+1}$ have a computational cost of $O(md)$, and the operations with H_t have a cost of $O(m^2)$. Hence, as long as we can maintain H_t and S_t efficiently, this new "Sketched MetaGrad" is then (much) faster than the original MetaGrad.

We show, in the following, ways of obtaining possible S_t matrices (and we introduce the corresponding modified versions of the matrix A_t in the MetaGrad slaves algorithm), based on random projections, and on "frequent directions", and study the impact of this sketching on the bounds of MetaGrad. Other approaches exist, for instance sparsifying matrices (cf. Arora et al. [2006] and Achlioptas and McSherry [2007]), or using only a subset of the columns or rows –"(sub)sampling"– of the matrices (cf. Boutsidis et al. [2009] and Boutsidis et al. [2014]).

4.2.3. Random Sketching

General principles. The first approach we adapt to MetaGrad is a "Random Sketching" presented in Luo et al. [2016], which is based on a multiplication by a random matrix. This idea of random dimensionality reduction can be found in Achlioptas [2001], or in Har-Peled et al. [2012] for the approximate nearest neighbours problem.

Following Luo et al. [2016], instead of appending a new row $\sqrt{2}\eta g_t^\top$ to G_{t-1} to update it into G_t, here we add to S_{t-1} the outer product $\sqrt{2}\eta r_t g_t^\top$ of $\sqrt{2}\eta g_t$ with a Gaussian random vector r_t of dimension m. The idea behind that approach is that with high probability, it achieves some kind of approximate "isometry" (cf. Theorem 2.3 of Woodruff [2014], and (4.2.1) below). We define the following update for S_t: $S_t = S_{t-1} + \sqrt{2}\eta r_t g_t^\top$ with $r_t \sim \mathcal{N}(0, (1/\sqrt{m})I_m)$. Then one can check that it is possible to update H_t^{-1} by two rank-one updates: $H_t^{-1} = H_{t-1}^{-1} + q_t r_t^\top + r_t q_t^\top$ where $q_t := \sqrt{2}\eta S_t g_t - \eta^2 \|g_t\|_2^2 r_t$. Therefore, using the Woodbury formula, this leads to a $O(md)$ update of S_t and H_t.

The sketching algorithm. The Random Sketching algorithm is described in Algorithm 10.

Algorithm 10 Random projection sketch

Input: dimension m.

Initialization:

1. Set $S_0 = \mathbf{0}_{m\times d}$.
2. Set $H_0 = \frac{1}{\varepsilon} I_m$.

Sketch update:

1. Draw $r_t \sim \mathcal{N}\left(0, \frac{1}{\sqrt{m}} I_m\right)$ and update $S_t = S_{t-1} + \sqrt{2}\eta r_t g_t^\top$.
2. Compute $q_t = \sqrt{2}\eta S_t g_t - \eta^2 \|g_t\|_2^2 r_t$.
3. Update $\tilde{H}_{t-1} = H_{t-1} - \frac{H_{t-1} q_t r_t^T H_{t-1}}{1 + r_t^T H_{t-1} q_t}$ and $H_t = \tilde{H}_{t-1} - \frac{\tilde{H}_{t-1} r_t q_t^T \tilde{H}_{t-1}}{1 + q_t^T \tilde{H}_{t-1} r_t}$.

Bounds obtained for MetaGrad. One has the following bounds for the η slave of the "Random Sketched MetaGrad". We recall that $\tilde{g}_t := \left(1 + 2\eta g_t^\top (w_t^\eta - w_t)\right) g_t$, and use an

adapted version of the quantities R_G and R_D defined in (4.1.4). We use a modified version of A_t that we denote by $A_{t,r}$ (r standing for "random projection").

Theorem 4.6. *Let $A_{t,r} = \varepsilon I_d + S_t^\top S_t$ and $\eta \in (0, \frac{1}{5GD}]$. For any $\delta \in (0,1)$ and any $\rho \in (0,1)$, if the sketch size m is such that $m = \Omega((rank(G_T) + \log(T/\delta))\rho^{-2})$ then the following holds. First, with probability greater than $1-\delta$, for any vector $u \in \mathcal{K}$:*

$$R_G := \sum_{t=1}^T \frac{\eta^2}{2} \tilde{g}_t^T A_{t,r}^{-1} \tilde{g}_t \leqslant \frac{1}{2(1-\rho)} \log\det\left(I_d + \frac{2\eta^2}{\varepsilon}\sum_{t=1}^T g_t g_t^\top\right)$$

Secondly, for any vector $u \in \mathcal{K}$:

$$R_D := \mathbb{E}\left[\sum_{t=1}^T \frac{1}{2}(u-w_t^\eta)^\top A_{t,r}(u-w_t^\eta) - \frac{1}{2}(u-w_{t+1}^\eta)^\top A_{t,r}(u-w_{t+1}^\eta) - \eta^2(u-w_t^\eta)^\top M_t(u-w_t^\eta)\right]$$
$$\leqslant \frac{\varepsilon}{2}\|u-w_1\|_2^2$$

Quite naturally, one has to make a trade-off between on the one hand the regret bound (ρ parameter) and the probability (δ parameter) of the algorithm, on the other hand, the size of the sketch m (which depends on δ and ρ). But if m if chosen correctly (i.e. large enough), then one can see by a comparison of Theorem 4.6 with (4.1.5) and (4.1.6), that the bounds of the original MetaGrad are preserved (exactly for R_D, up to a multiplicative constant and with probability $1-\delta$ for R_G).

One can easily see that the same bound remains valid if one replaces the domain $(0, 1/(5DG))$ by $(0, 1/(10LC))$ (where L and C are defined in the framework of Section 4.2.1), because one keeps then the bounds $3/5 \leqslant 1 + 2\eta g_t^\top(w_t^\eta - w_t) \leqslant 7/5$.

The fact that the bound on R_G holds with probability $1-\delta$ whereas for R_D we have a bound on an expectation, prevents uniting them to get a global expression on the regret (cf. decomposition (4.1.4)). In the following section, we will present another algorithm, "Frequent Directions", for which we will derive a complete regret bound: see (4.2.4).

Proof. The starting point is the following property of the random projection method (cf. Theorem 2.3 of Woodruff [2014]): since the sketch size m is such that $m = \Omega\left((rank(G_t) + \log(T/\delta))\,\rho^{-2}\right)$, one has on an event of probability greater than $1-\delta$:

$$\forall\, t = 1..T,\ (1-\rho)G_t^\top G_t \preceq S_t^\top S_t \preceq (1+\rho)G_t^\top G_t \tag{4.2.1}$$

(where $A \preceq B$ means that the matrix $B - A$ is positive semi-definite).

On this event, one has:

$$A_{t,r}^{-1} = \left(\varepsilon I_d + S_t^\top S_t\right)^{-1} \preceq \frac{1}{1-\rho}\left(\varepsilon I_d + G_t^\top G_t\right)^{-1} = \frac{1}{1-\rho}A_t^{-1}. \tag{4.2.2}$$

Combining the end of the proof of Lemma 4.3 with (4.2.2) leads to:

$$R_G := \sum_{t=1}^T \frac{\eta^2}{2}\tilde{g}_t^\top A_{t,r}^{-1}\tilde{g}_t \leqslant \frac{1}{2}\sum_{t=1}^T 2\eta^2 g_t^\top A_{t,r}^{-1} g_t \leqslant \frac{1}{2(1-\rho)}\log\left(\det\left(I_d + \frac{2\eta^2}{\varepsilon}\sum_{t=1}^T g_t g_t^\top\right)\right)$$

Let us tackle R_D now.

Since for $t \geqslant 1$, $A_{t,r} = \varepsilon I_d + S_t^\top S_t = \varepsilon I_d + \left(S_{t-1} + \sqrt{2}\eta r_t g_t\right)^\top \left(S_{t-1} + \sqrt{2}\eta r_t g_t\right)$, one has that for $t \geqslant 1$:

$$A_{t,r} - A_{t-1,r} = \sqrt{2}\eta S_{t-1}^\top r_t g_t^\top + \sqrt{2}\eta g_t r_t^\top S_{t-1} + 2\eta^2 \|r_t\|_2^2 g_t g_t^\top.$$

As r_t is independent of S_{t-1} and g_t, and $r_t \sim \mathcal{N}(0, (1/\sqrt{m})I_m)$, one has $\mathbb{E}\left[S_{t-1}^\top r_t g_t^\top\right] = 0$, $\mathbb{E}\left[g_t r_t^\top S_{t-1}\right] = 0$ and $\mathbb{E}\left[\|r_t\|_2^2 g_t g_t^\top\right] = \mathbb{E}\left[g_t g_t^\top\right] = \mathbb{E}\left[M_t\right]$. So $\mathbb{E}\left[A_{t,r} - A_{t-1,r}\right] = 2\eta^2 \mathbb{E}\left[M_t\right]$. But r_t is also independent of w_t^η, so one has for any $u \in \mathcal{K}$:

$$\mathbb{E}\left[(u - w_t^\eta)^\top \left(A_{t,r} - A_{t-1,r}\right)(u - w_t^\eta)\right] = 2\eta^2 \mathbb{E}\left[(u - w_t^\eta)^\top M_t (u - w_t^\eta)\right]$$

Therefore, R_D is a telescopic sum, and (using as in (4.1.5) w_{T+1} as an artifact of computations that does not need to be known):

$$\begin{aligned} R_D &= \mathbb{E}\left[\frac{1}{2}(u - w_1^\eta)^\top A_{1,r}(u - w_1^\eta) - \eta^2 (u - w_1^\eta)^\top M_1 (u - w_1^\eta) - \frac{1}{2}(u - w_{T+1}^\eta)^\top A_{T,r}(u - w_{T+1}^\eta)\right] \\ &\leqslant (u - w_1^\eta)^\top \mathbb{E}\left[\frac{1}{2}A_{1,r} - \eta^2 M_1\right](u - w_1^\eta) \\ &= \frac{\varepsilon}{2}\|u - w_1\|_2^2 \end{aligned}$$

■

4.2.4. Frequent directions algorithm

General principles. The algorithm we address in this section is deterministic. It is called "Frequent Directions" and is presented in Ghashami et al. [2016] and Liberty [2013]. We transfer some results and bounds obtained by Luo et al. [2016] (Theorem 3) for Online Newton Step, to MetaGrad.

The approach is based on a SVD decomposition of the sketch, followed by a deletion of its last row, replaced later by the new data (the vector $\sqrt{2\eta}g_t$). The SVD decomposition allows the procedure to lead to a matrix H_t that is diagonal and therefore easy to deal with.

The idea of deletion was originally used to compute item frequencies (cf. Misra and Gries [1982]). A collection of items can be represented as a Boolean matrix, whose columns form a dictionary and whose rows are indicator vectors. Then matrix sketching is the equivalent of"smart" count of items in Misra and Gries [1982]. The idea of deleting items so that only the most frequent ones remain, transposes as shrinking rows (and even making null at least one row at each step). Thus, one keeps only the most "frequent directions". This argument is presented in Ghashami et al. [2016].

The sketching algorithm. The "Frequent Directions" sketching algorithm is described in Algorithm 11.

As before, the matrices H_t and S_t built in the algorithm will be used to compute a modified version of A_t (and its inverse), that we will denote by $A_{t,\text{fd}}$ (fd standing for "frequent directions") and that will replace A_t in the slaves algorithm of MetaGrad.

Algorithm 11 Frequent Directions sketch

Input: dimension m.

Initialization:

1. Set $S_0 = \mathbf{0}_{m\times d}$.
2. Set $H_0 = \frac{1}{\varepsilon} I_m$.

Sketch update:

1. Replace the last row of S_{t-1} by $\sqrt{2}\eta g_t^\top$.
2. Compute eigendecomposition $S_{t-1}^\top S_{t-1} = V_t^\top \Sigma V_t$, with Σ diagonal of size $m \times m$, and V_t of size $m \times d$ satisfying: $V_t V_t^\top = I_m$.
3. Set $S_t = (\Sigma - \rho_t I_m)^{\frac{1}{2}} V_t$, where $\rho_t = \Sigma_{m,m}$.
4. Set $H_t = \operatorname{diag}\left(\frac{1}{\varepsilon+\Sigma_{1,1}-\rho_t}, \frac{1}{\varepsilon+\Sigma_{2,2}-\rho_t} \ldots, \frac{1}{\varepsilon}\right)$.

One can notice that in Algorithm 11, the last row of S_t is null, before being replaced by $\sqrt{2}\eta g_t^\top$ in Step 1 of the sketch update. As for the last diagonal term of H_t, it is $1/(\varepsilon+\Sigma_{m,m}-\rho_t)$ which is $1/\varepsilon$ by definition of ρ_t.

To help understanding, we recall here the dimensions of some of the vector and matrices at stake: g_t: $d \times 1$; S_t: $m \times d$ (so $S_t^\top S_t$: $d \times d$); G_t: $t \times d$; H_t: $m \times m$; V_t: $m \times d$; Σ: $m \times m$ (which is actually a bloc extracted from the full diagonal $d \times d$ matrix that appears in the eigendecomposition, dropping the last $d - m$ null eigenvalues).

Bounds obtained for MetaGrad. We use the same notations as in the "Random Sketching" section. We denote by $\lambda_1(G_T^\top G_T), \ldots, \lambda_d(G_T^\top G_T)$ the eigenvalues of the matrix $G_T^\top G_T$ sorted by decreasing order, and by $\Omega_k := \sum_{i=k+1}^{d} \lambda_i(G_T^\top G_T)$ the sum of the k smallest of them. Then, one has the following bounds for the η slave of the "Frequent Directions Sketched MetaGrad".

Theorem 4.7. *Let $A_{t,\mathrm{fd}} = \varepsilon I_d + S_t^T S_t$ and $\eta \in (0, \frac{1}{5GD}]$. Then the following holds for any $k < m$.*

$$
\begin{aligned}
R_G :=& \sum_{t=1}^{T} \frac{\eta^2}{2} \tilde{g}_t^\top A_{t,\mathrm{fd}}^{-1} \tilde{g}_t \leqslant \frac{1}{2}\left(\frac{m\Omega_k}{\varepsilon(m-k)} + \log(\det(I_d + \frac{1}{\varepsilon} S_T^\top S_T))\right) \\
R_D :=& \sum_{t=1}^{T} \frac{1}{2}(u - w_t^\eta)^\top A_{t,\mathrm{fd}}(u - w_t^\eta) - \frac{1}{2}(u - w_{t+1}^\eta)^\top A_{t,\mathrm{fd}}(u - w_{t+1}^\eta) - \\
& \qquad\qquad \eta^2 (u - w_t^\eta)^\top M_t (u - w_t^\eta) \\
\leqslant& \frac{\varepsilon}{2}\|u - w_1\|_2^2
\end{aligned}
\tag{4.2.3}
$$

Therefore, the following bound holds for the Slave algorithm:

$$\sum_{t=1}^{T} \ell_t^\eta(w_t^\eta) \leqslant \sum_{t=1}^{T} \ell_t^\eta(u) + \frac{\varepsilon}{2}\|u - w_1^\eta\|_2^2 + \frac{1}{2}\left(\frac{m\Omega_k}{\varepsilon(m-k)} + \log(\det(I_d + \frac{1}{\varepsilon}S_T^\top S_T))\right) \tag{4.2.4}$$

for any $u \in \mathcal{K}$.

One asset of this "frequent directions" approach is that it is deterministic and does not require the knowledge of $rank(G_T)$. The bound on R_G gives several guarantees simultaneously for various values of k.

Proof. For R_D, one first sees that $A_{t,\text{fd}} - A_{t-1,\text{fd}} = S_t^\top S_t - S_{t-1}^\top S_{t-1}$. Due to the sketch construction, if one denote S'_{t-1} the modified version of S_{t-1} with $\sqrt{2}\eta g_t$ inserted in the last row (replacing a null row), then $S_{t-1}'^\top S'_{t-1} = S_t^\top S_t + \rho_t V_t^\top V_t$ and a direct computation gives that $S_{t-1}'^\top S'_{t-1} = S_{t-1}^\top S_{t-1} + 2\eta^2 g_t g_t^\top$. So:

$$A_{t,\text{fd}} - A_{t-1,\text{fd}} = S_t^\top S_t - S_{t-1}^\top S_{t-1} = 2\eta^2 g_t g_t^\top - \rho_t V_t^T V_t \preceq 2\eta^2 g_t g_t^\top = 2\eta^2 M_t. \tag{4.2.5}$$

Therefore:

$$\frac{1}{2}(u - w_t^\eta)^\top A_{t,\text{fd}}(u - w_t^\eta) - \frac{1}{2}(u - w_t^\eta)^\top A_{t-1,\text{fd}}(u - w_t^\eta) \leqslant \eta^2 (u - w_t^\eta)^\top M_t (u - w_t^\eta)$$

So one can use the same argument as in previous Sections (with a telescopic inequality sum instead of a normal telescopic sum) to get that:

$$\begin{aligned}
R_D \leqslant \sum_{t=1}^{T-1} \Bigg(& \frac{1}{2}(u - w_t^\eta)^\top A_{t,\text{fd}}(u - w_t^\eta) - \frac{1}{2}(u - w_{t+1}^\eta)^\top A_{t+1,\text{fd}}(u - w_{t+1}^\eta) + \\
& \eta^2 (u - w_{t+1}^\eta)^\top M_{t+1}(u - w_{t+1}^\eta) - \eta^2 (u - w_t^\eta)^\top M_t (u - w_t^\eta)\Bigg) + \\
& \frac{1}{2}(u - w_T^\eta)^\top A_{T,\text{fd}}(u - w_T^\eta) - \frac{1}{2}(u - w_{T+1}^\eta)^\top A_{T,\text{fd}}(u - w_{T+1}^\eta) - \\
& \eta^2 (u - w_T^\eta)^\top M_T (u - w_T^\eta) \\
\leqslant & \frac{1}{2}(u - w_1^\eta)^\top A_{1,\text{fd}}(u - w_1^\eta) - \eta^2 (u - w_1^\eta)^\top M_1 (u - w_1^\eta) - \frac{1}{2}(u - w_{T+1}^\eta)^\top A_{T,\text{fd}}(u - w_{T+1}^\eta) \\
\leqslant & (u - w_1^\eta)^\top \left(\frac{1}{2}A_{1,\text{fd}} - \eta^2 M_1\right)(u - w_1^\eta) \\
= & \frac{\varepsilon}{2}\|u - w_1\|_2^2
\end{aligned}$$

Let us address R_G now.

As previously, since $\eta \in (0, 1/(5DG)]$, one has that

$$\frac{3}{5} \leqslant 1 + 2\eta g_t^\top (w_t^\eta - w_t) \leqslant \frac{7}{5}$$

and so:

$$\frac{\eta^2}{2}\tilde{g}_t^\top A_{t,\text{fd}}^{-1}\tilde{g}_t \leqslant \left(\frac{7}{5}\right)^2 \frac{\eta^2}{2} g_t^\top A_{t,\text{fd}}^{-1} g_t \leqslant \eta^2 g_t^\top A_{t,\text{fd}}^{-1} g_t. \tag{4.2.6}$$

Since $g_t^\top A_{t,\text{fd}}^{-1} g_t$ is a scalar, one has (with Tr denoting the trace of a matrix):

$$g_t^\top A_{t,\text{fd}}^{-1} g_t = \text{Tr}\left(g_t^\top A_{t,\text{fd}}^{-1} g_t\right) = \text{Tr}\left(A_{t,\text{fd}}^{-1} g_t g_t^\top\right) = \frac{1}{2}\text{Tr}\left(A_{t,\text{fd}}^{-1}(A_{t,\text{fd}} - A_{t-1,\text{fd}} + \rho_t V_t^T V_t)\right) \tag{4.2.7}$$

The second equality is an elementary property of the trace ($\text{Tr}(AB) = \text{Tr}(BA)$), and the last one comes from (4.2.5).

Then, we split the right-hand side of the previous equality into two parts.

As $A_{t,\text{fd}}^{-1} \preceq (1/\varepsilon) I_d$ and $V_t V_t^\top = I_m$, one has that:

$$\text{Tr}(A_{t,\text{fd}}^{-1} V_t V_t^\top) \leqslant \frac{1}{\varepsilon}\text{Tr}(V_t V_t^\top) = \frac{m}{\varepsilon}. \tag{4.2.8}$$

We will now use the following bound (cf. Theorem 1.1 of Ghashami et al. [2016]) on the eigenvalues that are "deleted" by the Frequent Directions sketching:

$$\forall\ k < m,\ \sum_{t=1}^{T} \rho_t \leqslant \frac{\Omega_k}{m-k}.$$

Combining this result and (4.2.8) gives:

$$\sum_{t=1}^{T} \text{Tr}(\rho_t A_{t,\text{fd}}^{-1} V_t V_t^\top) \leqslant \frac{m\Omega_k}{\varepsilon(m-k)}. \tag{4.2.9}$$

As for $\text{Tr}\left(A_{t,\text{fd}}^{-1}(A_{t,\text{fd}} - A_{t-1,\text{fd}})\right)$, we will use the concavity on symmetric positive definite matrices of the function $\log(\det(.))$, the fact that its gradient at a matrix A is A^{-1} (cf. Boyd and Vandenberghe [2004]), and we recall that the corresponding inner product is: $\langle A, B\rangle = \text{Tr}\,(AB)$. Thus, the concavity gradient inequality reads:

$$\text{Tr}\left(A_{t,\text{fd}}^{-1}(A_{t,\text{fd}} - A_{t-1,\text{fd}})\right) \leqslant \log\left(\det(A_{t,\text{fd}})\right) - \log\left(\det(A_{t-1,\text{fd}})\right).$$

This leads to:

$$\sum_{t=1}^{T} \text{Tr}\left(A_{t,\text{fd}}^{-1}(A_{t,\text{fd}} - A_{t-1,\text{fd}})\right) \leqslant \log\left(\frac{\det(A_{T,\text{fd}})}{\det(A_{0,\text{fd}})}\right) \leqslant \log(\det(I_d + \frac{1}{\varepsilon} S_T^\top S_T)). \tag{4.2.10}$$

Combining (4.2.6), (4.2.7), (4.2.9) and (4.2.10) gives the desired result:

$$R_G := \sum_{t=1}^{T} \frac{\eta^2}{2}\tilde{g}_t^\top A_{t,\text{fd}}^{-1}\tilde{g}_t \leqslant \frac{1}{2}\left(\frac{m\Omega_k}{\varepsilon(m-k)} + \log(\det(I_d + \frac{1}{\varepsilon} S_T^\top S_T))\right).$$

■

4.3. Improved "online-to-batch" conversions for Online Newton Step and MetaGrad

In the previous Section, we considered the Online Convex Optimization framework, dealing with individual deterministic sequences and getting uniform deterministic bounds (except for the Random Sketching part, where some randomness was introduced). In this Section, we switch towards a probabilistic "batch" framework, where the learner observes directly a whole learning sample of i.i.d. losses, and is evaluated on only one attempt.

In the introductory Chapter 2, we presented a "standard" way to convert individual sequences algorithm to the batch setting. In this Section we switch to more sophisticated approaches. We investigate which rates can be obtained by combining the techniques of Hazan and Kale [2014] (applied in their paper to Gradient Descent) with Online Newton Step and MetaGrad. Since these two algorithms obtain optimal (up to multiplicative constants) regret bounds $O(d \log T)$ for exp-concave losses in the online adversarial setting, our hope is that they will lead to optimal rates for exp-concave losses in the batch setting as well. Here we take a first step towards showing such a result by instead applying ONS and MetaGrad to strongly convex losses. This simplifies the analysis, because we stay closer to the setting of Hazan and Kale [2014], but since the regret bounds for ONS and MetaGrad on strongly convex losses are suboptimal by a factor d, we obviously also lose this factor in the batch rates that we obtain.

4.3.1. Framework

First-order information stochastic batch setting. We consider the batch setting of stochastic convex optimization. In this setting, we face i.i.d. convex loss functions $f_t, t \in 1, \dots, T$ on a convex domain $\mathcal{K} \subset \mathbb{R}^d$, and the goal is to minimize $F : x \in \mathcal{K} \mapsto \mathbb{E}[f_1(x)]$. Therefore, we want to mimic the performance of:

$$x^* = \operatorname*{argmin}_{x \in \mathcal{K}} F(x).$$

(we will assume that such x^* exists).

Before each time step t, we choose a point $x_t \in \mathcal{K}$. For any t, we do not have access directly to $f_t(x_t)$ (nor, a fortiori, to $F(x_t)$) but only to its (sub)gradient $g_t := \nabla f_t(x_t) \in \partial f_t(x_t)$ –we will assume that such a subgradient exist. This is a "noisy version" of $\nabla F(x_t)$.

Assumptions. We will assume that the subgradients of f_t are bounded by G, and that the diameter of the domain is not greater than D. Therefore, one has, for all $f_t, x_1 \in \mathcal{K}, x_2 \in \mathcal{K}$, that $f_t(x_2) - f_t(x_1) \leqslant \nabla f_t(x_2)^\top (x_2 - x_1)$ and $f_t(x_1) - f_t(x_2) \leqslant \nabla f_t(x_1)^\top (x_1 - x_2)$ so that the Cauchy-Schwarz inequality ensures that $|f_t(x_2) - f_t(x_1)| \leqslant G\|x_2 - x_1\|_2 \leqslant GD$. Therefore, by Jensen's inequality, $|F(x_1) - F(x_2)| \leqslant \mathbb{E}[|f_t(x_2) - f_t(x_1)|] \leqslant GD$.

We make an assumption stronger than convexity: we assume that all functions f_t are λ-strongly convex, which means that:

$$\forall x, y \in \mathcal{K}, \quad \forall\ \nabla f_t(y) \in \partial f_t(y), \quad f_t(x) \geqslant f_t(y) + \nabla f_t(y)^\top (x - y) + \frac{\lambda}{2}\|x - y\|_2^2$$

In particular, as $\left(\nabla f_t(y)^\top (x-y)\right)^2 \leqslant G^2\|x-y\|_2^2$ (via the Cauchy-Schwarz inequality), one has:

$$\forall x, y \in \mathcal{K},\ \forall\ \nabla f_t(y) \in \partial f_t(y),\ f_t(x) \geqslant f_t(y) + \nabla f_t(y)^\top (x-y) + \frac{\lambda}{2G^2}\left(\nabla f_t(y)^\top (x-y)\right)^2$$

Reversing this expression will allow us to bound the regret:

$$\forall x, y \in \mathcal{K},\ \forall\ \nabla f_t(y) \in \partial f_t(y),\ f_t(y) - f_t(x) \leqslant \nabla f_t(y)^\top (y-x) - \frac{\lambda}{2G^2}\left(\nabla f_t(y)^\top (x-y)\right)^2 \tag{4.3.1}$$

We will assume that F is also λ-strongly convex.

Some consequences of strong convexity. One can see that if the f_t are differentiable everywhere, then the $\lambda-$strong convexity of F is a direct consequence of the λ-strong convexity of the f_t. Indeed, in this case the boundedness of the gradients guarantees that F is also differentiable and allows to exchange gradient and expectation, by writing that $\mathbb{E}\left[\nabla f_t(y)\right] = \nabla \mathbb{E}\left[f_t(y)\right] = \nabla F(y)$ for any y and t. Therefore, one has:

$$\begin{aligned} F(x) - F(y) - \nabla F(y)^\top (x-y) &= \mathbb{E}\left[f_1(x) - f_1(y) - \nabla f_1(y)^\top (x-y)\right] \\ &\geqslant \mathbb{E}\left[\frac{\lambda}{2}\|x-y\|_2^2\right] \\ &= \frac{\lambda}{2}\|x-y\|_2^2 \end{aligned}$$

(Notice that in the general case, expectation of subgradients is trickier; in particular it is unclear, when a random function f_t can have a subgradient with multiple elements in a point y, what $\mathbb{E}\left[\nabla f_t(y)\right]$ means).

Another consequence of strong convexity with bounded gradient, is that it limits the domain. Consider two points x and y in $\mathcal{K}$ such that f_t has a non-empty subgradient in x and in y, and such that $\|x-y\|_2 \geqslant D/2$ (we assume that two such points exist). Then $\forall\ \nabla f_t(y) \in \partial f_t(y),\ \nabla f_t(y)^\top (y-x) \leqslant G\|x-y\|_2$, and similarly for x. So, applying the definition of the λ-strong convexity to x and y gives that $\lambda\|x-y\|_2^2/2$ is not greater than $f_t(x) - f_t(y) + G\|x-y\|_2$ and than $f_t(y) - f_t(x) + G\|x-y\|_2$. Therefore, $\lambda\|x-y\|_2^2/2 \leqslant G\|x-y\|_2$, so $\|x-y\|_2 \leqslant 2G/\lambda$, and since $\|x-y\|_2 \geqslant D/2$, one has $D \leqslant 4G/\lambda$. Conversely, knowing D and G enables to upper bound λ by $4G/D$.

Outline of the section. We will present "online-to-batch" conversions of two algorithms, we call these conversions "Epoch Online Newton Step" and "Epoch MetaGrad". "Epoch Online Newton Step" is more straightforward (and so is its analysis), but it requires to tune the learning rate η correctly, which requires the knowledge of the parameter λ of strong convexity. On the contrary, "Epoch MetaGrad" benefits from one of the main assets of MetaGrad: the learning rate η does not need to be tuned (and thus no knowledge about λ is required). The order of magnitude of the bounds is the same as for "Epoch Online Newton Step", up to a $O(\log\log(T))$ factor and a worsening of the constants.

4.3.2. Epoch Online Newton Step

Deterministic analysis

Let us first begin by an analysis of "classical" Online Newton Step (Algorithm 7, introduced in Hazan et al. [2007]). It follows the same line as the proof of Lemma 4.3, since the slave algorithm in MetaGrad is directly inspired by Online Newton Step.

The values of the parameters $\eta > 0$, $m > 0$ and $\varepsilon > 0$ will be fixed later.

Lemma 4.8. *Starting from an arbitrary point $x_1 \in \mathcal{K}$, and $A_1 = \varepsilon I_d + m g_1 g_1^\top$, apply T iterations of the update:*

$y_{t+1} = x_t - \eta A_t^{-1} g_t$
$x_{t+1} = \operatorname{argmin}_{x \in \mathcal{K}} (x - y_{t+1})^\top A_t (x - y_{t+1})$
$A_{t+1} = \varepsilon I_d + m \sum_{s=1}^{t+1} g_s g_s^\top$
Then for any point $x^* \in \mathcal{K}$, one has:

$$\sum_{t=1}^{T} \left(g_t^\top (x_t - x^*) - \frac{m}{2\eta} \left(g_t^\top (x_t - x^*) \right)^2 \right) \leqslant \frac{\eta d}{2m} \log(mG^2T/\varepsilon + 1) + \frac{\varepsilon}{2\eta} \|x_1 - x^*\|_2^2 \quad (4.3.2)$$

An important corollary is that for η and m such that $m/\eta \leqslant \lambda/G^2$, then by λ-strong convexity, as seen in (4.3.1), the left-hand side of (4.3.2) is an upper bound on the regret $\sum_{t=1}^T f_t(x_t) - f_t(x^*)$, and therefore the right-hand side of (4.3.2) too:

$$\sum_{t=1}^{T} f_t(x_t) - f_t(x^*) \leqslant \frac{\eta d}{2m} \log(mG^2T/\varepsilon + 1) + \frac{\varepsilon}{2\eta} \|x_1 - x^*\|_2^2 \quad (4.3.3)$$

Proof. $y_{t+1} - x^* = x_t - x^* - \eta_t A_t^{-1} g_t$ so

$$(y_{t+1} - x^*)^\top A_t (y_{t+1} - x^*) = (x_t - x^*)^\top A_t (x_t - x^*) - 2\eta g_t^\top (x_t - x^*) + \eta^2 g_t^\top A_t^{-1} g_t$$

A classical property of projections (cf. Hazan et al. [2007]) give that:
$(y_{t+1} - x^*)^\top A_t (y_{t+1} - x^*) \geqslant (x_{t+1} - x^*)^\top A_t (x_{t+1} - x^*)$. As a consequence,

$$g_t^\top (x_t - x^*) \leqslant \frac{1}{2} \left(\eta g_t^\top A_t^{-1} g_t + \frac{1}{\eta} (x_t - x^*)^\top A_t (x_t - x^*) - \frac{1}{\eta} (x_{t+1} - x^*)^\top A_t (x_{t+1} - x^*) \right)$$

Recalling that $A_{t+1} - A_t = m g_{t+1} g_{t+1}^\top$, one gets by an Abel's transform:

$$\begin{aligned}
\sum_{t=1}^{T} g_t^\top (x_t - x^*) &\leqslant \frac{\eta}{2} \sum_{t=1}^{T} g_t^\top A_t^{-1} g_t + \frac{1}{2\eta} (x_1 - x^*)^\top A_1 (x_1 - x^*) \\
&\quad + \sum_{t=2}^{T} \frac{1}{2\eta} (x_t - x^*)^\top (A_t - A_{t-1})(x_t - x^*) - \frac{1}{2\eta} (x_{T+1} - x^*)^\top A_T (x_{T+1} - x^*)) \\
&\leqslant \frac{\eta}{2} \sum_{t=1}^{T} g_t^\top A_t^{-1} g_t + \sum_{t=1}^{T} \frac{m}{2\eta} (x_t - x^*)^\top (g_t g_t^\top)(x_t - x^*) + \frac{\varepsilon}{2\eta} \|x_1 - x^*\|_2^2
\end{aligned}$$

From Lemma 4.4 (applied with $u_t = \sqrt{m} g_t$, and writing $g_t^\top A_t^{-1} g_t$ as $\sqrt{m} g_t^\top A_t^{-1} \sqrt{m} g_t / m$) one gets that $\sum_{t=1}^T g_t^\top A_t^{-1} g_t \leqslant (d/m) \log(mG^2 T/\varepsilon + 1)$.

As a consequence, one has:

$$\sum_{t=1}^T \left(g_t^\top (x_t - x^*) - \frac{m}{2\eta} (x_t - x^*)^\top (g_t g_t^\top)(x_t - x^*) \right) \leqslant \frac{\eta d}{2m} \log(mG^2 T/\varepsilon + 1) + \frac{\varepsilon}{2\eta} \|x_1 - x^*\|_2^2$$

■

If one takes $\varepsilon = T$, then the right-hand side of the previous inequality is equal to $(\eta d/2m) \log(mG^2 + 1) + T\|x_1 - x^*\|_2^2/(2\eta)$. This will be key in our approach.

The algorithm

Classical "online-to-batch" conversion. The bound (4.3.3) would allow to use a classical "online-to-batch" conversion (cf. Section 2.3.3): starting from an arbitrary point x_1, run on the whole learning sample ("as if it were online") the version of Online Newton Step described in Lemma 4.8, and then output $\bar{x} := \sum_{t=1}^T x_t/T$. Then:

$$\begin{aligned} \mathbb{E}[F(\bar{x})] - F(x^*) \leqslant \mathbb{E}\left[\frac{1}{T} \sum_{t=1}^T F(x_t) \right] - F(x^*) &= \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^T f_t(x_t) - f_t(x^*) \right] \\ &\leqslant \frac{\eta d \log(mG^2 T/\varepsilon + 1)}{2mT} + \frac{\varepsilon \|x_1 - x^*\|_2^2}{2\eta T} \end{aligned}$$

the first inequality coming from the convexity of F, and the last one using (4.3.2).
$\mathbb{E}[F(x_t)] = \mathbb{E}[f_t(x_t)]$ relies on the fact that f_t is independent of x_t, that depends only on $f_1, \ldots, f_{t-1}$ (which are random functions, so x_t is random, therefore $\mathbb{E}[F(x_t)]$ is indeed an expectation, even if F is deterministic).

In terms of d and T, this bound is $O(d \log(T)/T)$, which is suboptimal in the strongly convex set-up. We then detail now an improved algorithm, that we call "Epoch Online Newton Step", which achieves an accuracy of $O(d/T)$ as for d and T (unfortunately, it is still not the optimal rate, which is $O(1/T)$ for strongly convex losses).

Epoch Online Newton Step algorithm. The algorithm, and its analysis, are strongly inspired by Hazan and Kale [2014]. They apply their ideas to a "linear first-order" method (Gradient Descent) whereas we adapt them to a "quadratic first-order method": Online Newton Step.

The details are given in Algorithm 12.

We will define epochs of size T_j (for the j-th epoch), and we will use a double indexation in t and j: f_t^j, x_t^j, with t the "local" time inside epoch j, corresponding to a global time $(\sum_{i=1}^{j-1} T_i) + t$. In other words, we will write f_t^j and x_t^j, instead respectively of $f_{(\sum_{i=1}^{j-1} T_i)+t}$ and $x_{(\sum_{i=1}^{j-1} T_i)+t}$

The approach relies on the following key idea. The theoretical bounds on the average regret that one gets on running classical Online Newton Steps (derived from Lemma 4.8) decrease

with T, but increase with the initial distance to the objective: $\|x_1 - x^*\|_2$. Therefore, if the algorithm tends to converge fast enough to the objective (and the analysis will show that it is the case), then it is reasonable to make a trade-off between length of run T and accuracy of the starting point. Roughly, we "sacrifice" the first half of the learning sample to find a point close to the objective x^*, and we run Online Newton Step on the second half of the learning sample, initializing on this point "close to the objective".

As distance to the objective and performance are linked, the search for a point close to the objective (first half of the learning sample) leads us to seek a point with a good performance -that is why we iterate the previous idea in this first half of the learning sample. This leads us to some kind of dichotomy and to divide the learning sample into "epochs" whose size is exponentially increasing.

We present now the algorithm (Algorithm 12 below) with further details. We divide the time $t = 1 \dots, T$ in epochs in the following way. Choose T_1, and for any $j \in \mathbb{N}$, $T_j = 2^{j-1}T_1$. Let k be the number of completed epoch, i.e., such that $\sum_{i=1}^{k} T_i \leqslant T < \sum_{i=1}^{k+1} T_i$. We define the first epoch as $t = 1, \dots, T_1$ and the j-th epoch as $t = T_{j-1} + 1, \dots, T_j$ and we write $x_t^j := x_{(\sum_{i=1}^{j-1} T_i)+t}$.

We start in epoch 1, from an arbitrary point x_1^1. In epoch j, we run a version of the Online Newton Step algorithm with parameter η constant over the epochs, and parameter $\varepsilon_j = 2^j \varepsilon_0$, but we initialise it at the average of the outputs of the previous epoch: $x_1^j = \sum_{t=1}^{T_{j-1}} x_t^{j-1}/T_{j-1}$. This allows to decrease the upper bound on $\|x_1^j - x^*\|_2$ at each epoch.

Contrary to Hazan and Kale [2014], we keep a constant learning rate η. This comes from the presence of the quadratic term $m\left(g_t^\top (x_t - x^*)\right)^2/\eta$ in Lemma 4.8 and therefore in our subsequent analysis. To handle this term with the strong convexity property, we need to keep m/η small. Not too small though, because its inverse η/m appears in the right-hand side bound of Lemma 4.8. So we decide to keep η/m constant, and modify instead another parameter (which does not exist in the Epoch Gradient Descent of Hazan and Kale [2014]): the initialization $\varepsilon_j I_d$ of A_t^j.

The output x_1^{k+1} of the algorithm is the average of the outputs of the last epoch: $x_1^{k+1} = \sum_{t=1}^{T_k} x_t^k/T_k$.

Theoretical bounds

The main result of this section shows that the algorithm "Epoch Online Newton step" achieves (as far as the dependency in the dimension d and the learning sample size T are concerned) a regret $O(d/T)$. This achieves the optimal rate in T. But we were not able to suppress the d factor, that can be seen in the regret bound of Theorem 2 of Hazan et al. [2007] (in the weaker hypothesis of exp-concavity), but which is suboptimal in the context of strong convexity (cf. Hazan and Kale [2014]). Whether this d term is an artifact of our analysis or is intrinsic to our algorithm is an open question.

Theorem 4.9. *Initialise the algorithm "Epoch Online Newton Step" with* $\eta = 4/\lambda$, $m = \eta\lambda/G^2 = 4/G^2$, $T_1 = \lceil \frac{2Gd\log(4G^2+1)}{\lambda D} \rceil$, *and* $\varepsilon_0 = T_1/2$.

Algorithm 12 Epoch Online Newton Step

Input: parameters $\eta, m, \varepsilon_0, T_1$ and total time T, domain $\mathcal{K}$.

Initialization:

1. Initialize x_1^1 arbitrarily
2. Set $j = 1$

while $\sum_{i=1}^{j} T_i \leqslant T$ do:

Epoch j

1. $\varepsilon_j = 2\varepsilon_{j-1}$
2. $A_0^j = \varepsilon_j I_d$
3. **for** $t = 1$ to T_j do:
4. obtain $g_t^j \in \partial f_t^j(x_t^j)$ with $f_t^j := f_{(\sum_{i=1}^{j-1} T_i)+t}$

   ### Update:
5. $A_t^j = A_{t-1}^j + m g_t^j g_t^{j\top}$
6. $y_{t+1}^j = x_t^j - \eta \left(A_t^j\right)^{-1} g_t^j$
7. $x_{t+1}^j = \operatorname*{argmin}_{x \in \mathcal{K}} (x - y_{t+1}^j) A_t^j (x - y_{t+1}^j)$
8. **end for**
9. Set $x_1^{j+1} = \frac{1}{T_j} \sum_{t=1}^{T_j} x_t^j$
10. Set $T_{j+1} = 2T_j$
11. Set $j \leftarrow j + 1$

end while

Output: x_1^j *### If the number of completed epochs is k, this is x_1^{k+1}*

Then the output x_1^{k+1} of the algorithm satisfies:

$$\mathbb{E}[F(x_1^{k+1})] - F(x^*) \leqslant \frac{4G^2 d \log(4G^2+1)}{\lambda T} + \frac{2GD}{T}$$

This is a $O(d/T)$ bound.

One can notice that there exists also a trivial bound, independent of T and d, for $\mathbb{E}[F(x_1^k)] - F(x^*)$:

$$\mathbb{E}[F(x_1^{k+1})] - F(x^*) \leqslant GD \leqslant 4G^2/\lambda. \tag{4.3.4}$$

The first inequality comes from: $\mathbb{E}[F(x_1^{k+1})] - F(x^*) \leqslant \mathbb{E}[|F(x_1^{k+1}) - F(x^*)|] \leqslant GD$, and the second one is a direct consequence of strong convexity, already stated in Section 4.3.1.

Proof. By the hypothesis of λ-strong convexity of f, since $m/\eta = \lambda/G^2$, one has, similarly to (4.3.1):

$$\sum_{t=1}^{T_j} \left(f_t^j(x_t^j) - f_t^j(x^*) \right) \leqslant \sum_{t=1}^{T_j} \left(g_t^{j\top}(x_t^j - x^*) - \frac{m}{2\eta}(x_t^j - x^*)^\top (g_t^j g_t^{j\top})(x_t^j - x^*) \right)$$

and the right-hand-side can itself be upper bounded using Lemma 4.8. Since F is λ-strongly convex, the null vector is in $\partial F(x^*)$ (because $x^* \in \operatorname{argmin} F$), and the definition of λ-strong convexity leads to:

$$\|x_1 - x^*\|_2^2 \leqslant 2(F(x_1) - F(x^*))/\lambda. \tag{4.3.5}$$

Moreover, the independence of x_t^j and f_t^j gives that $\mathbb{E}[f_t^j(x_t^j)] = \mathbb{E}[F(x_t^j)]$ (which remains an expectation since x_t^j depends on the previous f_s). The same result holds for the expectation conditioned on all randomness until the end of the epoch $j-1$ (that we will write E_{j-1}): $E_{j-1}[f_t^j(x_t^j)] = E_{j-1}[F(x_t^j)]$. The convexity of the f_t, and therefore of F, gives that:

$$F\left(\frac{1}{T_j} \sum_{t=1}^{T_j} x_t^j \right) \leqslant \frac{1}{T_j} \sum_{t=1}^{T_j} F(x_t^j)$$

and this also holds in expectation.

The initialization of T_1 and ε_0 leads to: $\varepsilon_j = T_j$ for any $j \geqslant 1$.

One can now bring all these pieces together. Since $x_1^{j+1} = (\sum_{t=1}^{T_j} x_t^j)/T_j$ for any $j \geqslant 1$, one has by Lemma 4.8, conditioning on all randomness until the end of the $(j-1)$-th epoch:

$$\begin{aligned}
E_{j-1}[F(x_1^{j+1})] - F(x^*) &\leqslant \frac{1}{T_j} E_{j-1}\left[\sum_{t=1}^{T_j} F(x_t^j) - F(x^*) \right] \\
&= \frac{1}{T_j} E_{j-1}\left[\sum_{t=1}^{T_j} f_t^j(x_t^j) - f_t^j(x^*) \right] \\
&\leqslant \frac{\eta d \log(mG^2 T_j/\varepsilon_j + 1)}{2mT_j} + \frac{\varepsilon_j \left(F(x_1^j) - F(x^*) \right)}{\eta \lambda T_j} \\
&= \frac{\eta d \log(mG^2+1)}{2mT_j} + \frac{F(x_1^j) - F(x^*)}{\eta\lambda}.
\end{aligned}$$

Denote $\Delta_j := F(x_1^j) - F(x^*)$. Taking the unconditional expectation of the previous expression, and using the fact that $\eta\lambda = 4$, gives:

$$\mathbb{E}[\Delta_{j+1}] \leqslant \frac{U_\eta}{T_j} + \frac{\mathbb{E}[\Delta_j]}{4}$$

with $U_\eta = \eta d \log(mG^2+1)/2m = dG^2 \log(4G^2+1)/2\lambda$. Note that we picked $T_1 = \lceil 4U_\eta/(GD) \rceil$, which yields $U_\eta/T_1 \leqslant GD/4$.

Then, define $V_j = 2^{1-j}GD$.

Lemma 4.10. *For all $j \geqslant 1$, $\mathbb{E}[\Delta_j] \leqslant V_j$.*

Proof. Let us show it by induction. It is true for $j = 1$, because GD is a bound on $F(x_1^1) - F(x^*)$ so $\mathbb{E}[\Delta_1] \leqslant GD$. Let us assume that the property is true for some $j \geqslant 1$, and show that it is also true for $j+1$. One has:

$$\mathbb{E}[\Delta_{j+1}] \leqslant \frac{U_\eta}{T_j} + \frac{\mathbb{E}[\Delta_j]}{4} = \frac{U_\eta}{2^{j-1}T_1} + \frac{\mathbb{E}[\Delta_j]}{4} \leqslant \frac{2^{1-j}GD}{4} + \frac{V_j}{4} = V_{j+1}.$$

The first inequality has been seen before, and the second inequality relies on the fact that $T_1 = \lceil 4U_\eta/(GD) \rceil$ and on the induction hypothesis. This finishes to prove the lemma. ∎

One can then complete the proof of Theorem 4.9.

A straightforward computation gives that the number k of complete epochs run by the algorithm is: $k = \lfloor \log_2(1+T/T_1) \rfloor$. Moreover, we recall that $T_1 \leqslant (4U_\eta/B)+1$. Consequently, by Lemma 4.10, one has:

$$\mathbb{E}[F(x_1^{k+1})] - F(x^*) = \mathbb{E}[\Delta_{k+1}] \leqslant V_{k+1} = 2^{-k}GD \leqslant \frac{2T_1GD}{T} \leqslant \frac{8U_\eta + 2GD}{T}.$$

Since $U_\eta = d\,G^2 \log(4G^2+1)/2\lambda$, this gives the desired bound. ∎

4.3.3. Epoch MetaGrad

Deterministic analysis

We recall that $\tilde{R}_T^{x^*} := \sum_{t=1}^T g_t^\top (x_t - x^*)$ and that $V_T^{x^*} := \sum_{t=1}^T \left(g_t^\top (x_t - x^*)\right)^2$. We also recall the grid of learning rates in the MetaGrad master algorithm (Step 1 of Algorithm 8): $\eta_i = 2^{-i}/(5DG)$ for $i = 0, 1, \ldots, \lceil \log_2(T)/2 \rceil$ with prior weights $\pi_1^{\eta_i} = \frac{1+1/(1+\lceil \log_2(T)/2 \rceil)}{(i+1)(i+2)}$.

We re-write for convenience an adapted version of Lemma 4.5, denoting η_i instead of η to emphasize that several values of η_i, within the grid, are used by MetaGrad.

Lemma 4.11. *Starting from an arbitrary point $x_1 \in K$, apply T iterations of MetaGrad, using as covariance matrix initialization $A_0^{\eta_i} = \varepsilon_i I_d$. Then, for any value $\eta_i \in (0, \frac{1}{5DG}]$:*

$$\tilde{R}_T^{x^*} \leqslant \eta_i V_T^{x^*} + \frac{1}{\eta_i}\left(\frac{\varepsilon}{2}\|x_1 - x^*\|_2^2 - \log(\pi_1^{\eta_i}) + \frac{1}{2}\log\left(\det\left(I_d + 2\frac{\eta_i^2}{\varepsilon}\sum_{t=1}^T g_t g_t^\top\right)\right)\right) \quad (4.3.6)$$

An extra difficulty. Contrary to "Epoch Online Newton Step", here it is not directly possible to use (4.3.1) to write $\sum_{t=1}^T f_t(x_t) - f_t(x^*) \leqslant \tilde{R}_T^{x^*} - \eta_i V_T^{x^*}$ for all $\eta_i \leqslant 1/(5DG)$, it is only guaranteed for η_i smaller than $\lambda/(2G^2)$. To overcome this problem, we introduce the following quantity, that mixes the bounds required by (4.3.1) and by Lemma 4.11:

$$\zeta := \min\left(\lambda/(2G^2), 1/(5DG)\right).$$

Therefore, any $\eta_i \leqslant \zeta$ satisfies $\sum_{t=1}^T f_t(x_t) - f_t(x^*) \leqslant \tilde{R}_T^{x^*} - \eta_i V_T^{x^*}$ and Lemma 4.11.

This will allow us to prove the following result.

Lemma 4.12. *Defining:*

$$\xi_T := \min\left(2\log(4+\log_2(T)/2), 2\log\left(-\log_2(\zeta/2) - \log_2(5DG) + 2\right)\right)$$

one has:

$$\sum_{t=1}^T f_t(x_t) - f_t(x^*) \leqslant \tilde{R}_T^{x^*} - \zeta V_T^{x^*} \leqslant \frac{2}{\zeta}\left(\frac{\varepsilon}{2}\|x_1 - x^*\|_2^2 + \xi_T + \frac{d}{2}\log(1+T/(\varepsilon D^2))\right). \quad (4.3.7)$$

We will see that this quantity ξ_T is actually an upper bound on $-\log(\pi_1^{\eta_i})$ for an η_i carefully chosen (either close to ζ, or the smallest value in the grid).

Remark: we could have proved a similar lemma focused on the grid point closest to $\widehat{\eta}$ –that will be introduced in the proof in (4.3.10)–, instead of ζ; but we chose to introduce ζ to work with a fixed quantity in the following sections.

Proof. We will separate two cases, depending whether ζ is larger or smaller than the smallest point of the grid, that we will denote by η_ℓ (ℓ standing for low).

First case. First, assume that ζ is larger than η_ℓ (and by definition $\zeta \leqslant \eta_0 = 1/(5DG)$). Then there exists η_i in the grid such that $\zeta \in [\eta_i, 2\eta_i]$. Then, using $\pi_1^{\eta_i} \geqslant 1/(i+2)^2$ and $\eta_i = 2^{-i}/(5DG)$, one has:

$$\begin{aligned}
-\log(\pi_1^{\eta_i}) &\leqslant 2\log(i+2) \\
&= 2\log\left(-\log_2(\eta_i) - \log_2(5DG) + 2\right) \\
&\leqslant 2\log\left(-\log_2(\zeta/2) - \log_2(5DG) + 2\right)
\end{aligned}$$

We denote by ξ this last quantity:

$$\xi := 2\log\left(-\log_2(\zeta/2) - \log_2(5DG) + 2\right). \quad (4.3.8)$$

Notice that ξ is the second term in the definition of ξ_T.

One has $2\eta_i^2 \leqslant 2\zeta^2 < 1/(D^2G^2)$ since $\zeta \leqslant 1/(5DG)$, so one can upper bound $\log(\det(I_d + \frac{2\eta_i^2}{\varepsilon}\sum_{t=1}^T g_t g_t^\top))$ by $d\log(1+T/(\varepsilon D^2))$ (the greatest eigenvalue of $I_d + \frac{1}{\varepsilon G^2 D^2}\sum_{t=1}^T g_t g_t^\top$ can not be greater than $1 + (T/(\varepsilon D^2))$. As $\eta_i \leqslant \zeta$ and $1/\eta_i \leqslant 2/\zeta$, one has from (4.3.6) (and using the λ-strong convexity for the first inequality):

$$\sum_{t=1}^T f_t(x_t) - f_t(x^*) \leqslant \tilde{R}_T^{x^*} - \zeta V_T^{x^*} \leqslant \frac{2}{\zeta}\left(\frac{\varepsilon}{2}\|x_1 - x^*\|_2^2 + \xi + \frac{d}{2}\log(1+T/(\varepsilon D^2))\right)$$

Notice that since ζ is larger than η_ℓ, one has (due to the construction of the grid):

$$\frac{\zeta}{2} \geqslant \frac{\eta_\ell}{2} \geqslant \frac{1}{20DG\sqrt{T}}$$

and thus, using the definition (4.3.8) of ξ: $\xi \leqslant 2\log(4+\log_2(T)/2)$. Therefore, $\xi_T = \xi$ and the bound (4.3.7) is satisfied for this first case.

Second case. We will use similar arguments for the case in which ζ is smaller than the smallest grid point η_ℓ. Due to the construction of the grid, $2\eta_\ell^2 \leqslant 2\eta_0^2 \leqslant 1/(D^2G^2)$ and $-\log(\pi_1^{\eta_\ell}) \leqslant 2\log(3+\log_2(T)/2)$, so one can upper bound the right-hand side of (4.3.6), applied to η_ℓ:

$$\tilde{R}_T^{x^*} \leqslant \eta_\ell V_T^{x^*} + \frac{M}{\eta_\ell} \tag{4.3.9}$$

where:

$$M = \left(\frac{\varepsilon}{2}\|x_1 - x^*\|_2^2 + 2\log(3+\log_2(T)/2) + \frac{1}{2}\log\left(\det\left(I_d + \frac{1}{\varepsilon G^2 D^2}\sum_{t=1}^T g_t g_t^\top\right)\right)\right)$$

The minimizer $\widehat{\eta}$ of $\eta \mapsto \eta V_T^{x^*} + M/\eta$ is:

$$\widehat{\eta} = \sqrt{M/V_T^{x^*}} \geqslant \sqrt{2\log(3+\log_2(T)/2)/(TG^2D^2)} \geqslant 1/(5GD\sqrt{T}) \geqslant \eta_\ell \geqslant \zeta \tag{4.3.10}$$

By convexity of $\eta \mapsto \eta V_T^{x^*} + M/\eta$, one gets that $\eta_\ell V_T^{x^*} + M/\eta_\ell \leqslant \zeta V_T^{x^*} + M/\zeta$, so (4.3.9) gives: $\tilde{R}_T^{x^*} - \zeta V_T^{x^*} \leqslant M/\zeta$. The previously seen inequality $\log(\det(I_d + \frac{1}{\varepsilon G^2 D^2}\sum_{t=1}^T g_t g_t^\top)) \leqslant d\log(1+T/(\varepsilon D^2))$ gives finally:

$$\sum_{t=1}^T f_t(x_t) - f_t(x^*) \leqslant \tilde{R}_T^{x^*} - \zeta V_T^{x^*} \leqslant \frac{1}{\zeta}\left(\frac{\varepsilon}{2}\|x_1 - x^*\|_2^2 + 2\log(3+\log_2(T)/2) + \frac{d}{2}\log(1+T/(\varepsilon D^2))\right)$$

One can notice that in that case, since ζ is smaller than the smallest grid point η_ℓ, one has $\zeta/2 \leqslant 1/(10GD\sqrt{T})$, so (due to the definition of ξ) $2\log(3+\log_2(T)/2) \leqslant \xi$. Therefore, $2\log(3+\log_2(T)/2) \leqslant \xi_T$ and the bound (4.3.7) is satisfied in this second case. ∎

The algorithm

A classical "online-to-batch" conversion (running the original MetaGrad algorithm on the whole learning sample, "as if it were online", and then outputting the average of the forecasts, cf. Section 2.3.3) would lead to a $O(d\log(T)/T)$ upper bound on the expected regret (i.e., the bound (4.3.6) divided by T: cf. Theorem 2.9). To improve on this, we modify the conversion into an algorithm that we call Epoch MetaGrad (Algorithm 13).

We keep the same idea as in the Epoch Online Newton Step algorithm: dividing the learning sample into epochs, and running MetaGrad within each epoch, with an adequate choice of parameters and starting from the average of the outputs in the previous epoch. Thus, we can guarantee that at each epoch we get closer to the objective x^*, both in terms of distance $\|x_1^j - x^*\|$ and in terms of expected regret (more precisely, we get upper bounds on these two quantities that decrease quickly at each epoch).

Algorithm 13 Epoch MetaGrad

Input: total time T, domain $\mathcal{K}$, diameter of the domain D, uniform bound on the (sub-)gradients G.

Initialization:

1. Initialize x_1^1 arbitrarily.
2. Set $j = 1$, $T_1 = 2$ and $\varepsilon_1 = T_1/(2\log(T_1))$.

while $\sum_{i=1}^{j} T_i \leqslant T$ do:
Epoch j

1. Run T_j rounds of MetaGrad, initialized with $\varepsilon_j I_d$ and x_1^j, get the outputs x_t^j.
2. Set $x_1^{j+1} = \frac{1}{T_j}\sum_{t=1}^{T_j} x_t^j$
3. Set $T_{j+1} = 2T_j$
4. Set $\varepsilon_j = T_j/(2\log(T_j))$
5. Set $j \leftarrow j+1$

end while

Output: x_1^j *### If the number of completed epochs is k, this is* x_1^{k+1}

Theoretical bounds

Our main result shows that we get with Epoch MetaGrad nearly the same order of magnitude of expected error than for the Epoch Online Newton Step algorithm: $O(d\log\log(T)/T)$ (as far as d and T are concerned), but without having to tune η using an *a priori* knowledge on λ: **we gain adaptivity in** λ.

We focus on theses dependencies in d and T and might be a bit loose towards the other parameters.

We have previously introduced the following quantities, independent of d and (for T large enough) of T: $\zeta := \min\left(\lambda/(2G^2), 1/(5DG)\right)$ and

$$\xi_T := \min\left(2\log(4+\log_2(T)/2), 2\log\left(-\log_2(\zeta/2) - \log_2(5DG) + 2\right)\right).$$

Define now: $\rho_T := \lceil 16\xi_T/(GD\zeta)\rceil$.

One can easily see that ρ_T is constant for T large enough.
For T large enough, $2\log_2(4+\log_2(T)/2) \geqslant 2\log\left(-\log_2(\zeta/2) - \log_2(5DG) + 2\right)$ so $\xi_T = 2\log\left(-\log_2(\zeta/2) - \log_2(5DG) + 2\right)$ and $\rho_T = \lceil 16\log\left(-\log_2(\zeta/2) - \log_2(5DG) + 2\right)/(\zeta GD)\rceil$.

Theorem 4.13. *The output* x_1^{k+1} *of the "Epoch MetaGrad" algorithm satisfies:*

$$\mathbb{E}[F(x_1^{k+1})] - F(x^*) \leqslant \frac{8GD}{T}\max\left(\rho_T, \frac{8d}{GD\zeta}, \exp\left(\frac{4}{\zeta\lambda}\right)\right)\max\left(1, \log\left(1+\frac{2\log(T)}{D^2}\right)\right).$$

In particular, as far as the dependencies in d and T are concerned,

$$\mathbb{E}[F(x_1^{k+1})] - F(x^*) \underset{T\to\infty}{=} O(d \log\log(T)/T).$$

Similarly to (4.3.4), one has also the following trivial bound, independent of T and d:

$$\mathbb{E}[F(x_1^{k+1})] - F(x^*) \leqslant GD.$$

Proof. The proof follows the line of the proof for Epoch Online Newton Step (Theorem 4.9). The same arguments (independence of x_t^j and f_t^j, convexity) give:

$$\mathbb{E}[F(x_1^{j+1})] - F(x^*) \leqslant \frac{1}{T_j}\sum_{t=1}^{T_j} \mathbb{E}[f_t^j(x_t^j) - f_t^j(x^*)].$$

Using the fact, already seen in (4.3.5), that $\|x_1^j - x^*\|_2^2 \leqslant 2(F(x_1^j) - F(x^*))/\lambda$ (by strong convexity and since $0 \in \partial F(x^*)$, because x^* minimizes F), one then has, taking the expectation of (4.3.7):

$$\mathbb{E}[F(x_1^{j+1})] - F(x^*) \leqslant \frac{2}{\zeta T_j}\left(\frac{\varepsilon_j}{\lambda}\left(\mathbb{E}[F(x_1^j)] - F(x^*)\right) + \xi_{T_j} + \frac{d}{2}\log(1 + T_j/(\varepsilon_j D^2))\right) \quad (4.3.11)$$

We will for the moment assume that there exists a completed epoch j_1 satisfying:

$$T_{j_1} \geqslant \max\left(\rho_T, 8d/(GD\zeta), \exp\left(4/(\zeta\lambda)\right)\right) \quad (4.3.12)$$

We will start our analysis at the beginning of the first epoch satisfying the previous inequality, and re-number everything by defining this epoch as epoch 1, without loss of generality. The only impact is that the remaining learning sample is of size $\tilde{T} \geqslant T/2$ (and, of course, T_1 and ε_1 have no longer the value given in the description of the algorithm).

After this re-numbering, denote $\Delta_j = F(x_1^j) - F(x^*)$ and

$$V_j = 2^{1-j} GD \max\left(1, \log(1 + 2\log(T_j)/D^2)\right).$$

The same lemma as in section 4.3.2 holds (although the value of V_j is slightly different).

Lemma 4.14. *For all $j \geqslant 1$, $\mathbb{E}[\Delta_j] \leqslant V_j$.*

Proof. We show it by induction. It is clear for $j = 1$: GD is a bound on $f_t(x) - f_t(y)$ for any $x, y \in \mathcal{K}$, so $\mathbb{E}[\Delta_1] \leqslant GD \leqslant V_1$. Let us assume the property holds for some $j \geqslant 1$, and show that it is also true for $j+1$. From (4.3.11), one has:

$$\mathbb{E}[\Delta_{j+1}] \leqslant \frac{2\varepsilon_j}{T_j\zeta\lambda}\mathbb{E}[\Delta_j] + \frac{2\xi_{T_j}}{\zeta T_j} + \frac{d\log(1 + T_j/(\varepsilon_j D^2))}{\zeta T_j}$$

Therefore, given the value of ε_j and T_j (after renumbering, cf. (4.3.12)), one has:

- $2\varepsilon_j/(T_j\zeta\lambda) = 1/4$ (since $2\log(T_j) \geqslant 8/(\zeta\lambda)$),
- $2\xi_{T_j}/(\zeta T_j) \leqslant 2^{1-j}GD/8$ (since $T_{j_1} = \rho_T := \lceil 16\xi_T/(GD\zeta)\rceil$),
- $d\log(1+T_j/(\varepsilon_j D^2))/(\zeta T_j) \leqslant 2^{1-j}GD\log(1+2\log(T_j)/D^2)/8$ (since $T_{j_1} \geqslant 8d/(GD\zeta)$

So:

$$\mathbb{E}[\Delta_{j+1}] \leqslant \frac{V_j}{4} + \frac{2^{1-j}GD}{8} + \frac{2^{1-j}GD\log(1+2\log(T_j)/D^2)}{8} \leqslant \frac{V_j}{2} \leqslant V_{j+1}$$

This finishes to prove the lemma. ∎

We can now finish the theorem in the case where there exists an epoch satisfying (4.3.12). After renumbering, the number k of epoch run by the algorithm is: $k = \lfloor \log_2(1+\tilde{T}/T_1)\rfloor$, with $\tilde{T} \geqslant T/2$, so $2^{-k} \leqslant 4T_1/T$, with $T_1 \leqslant 2\max(\rho_T, 8d/(GD\zeta), \exp(4/(\zeta\lambda)))$. Another useful fact will be that $T_k \leqslant T/2$.

Consequently, by Lemma 4.14, one has:

$$\begin{aligned}
\mathbb{E}[F(x_1^{k+1})] - F(x^*) &= \mathbb{E}[\Delta_{k+1}] \\
&\leqslant V_{k+1} \\
&= 2^{-k}GD\ \max\left(1, \log(1+2\log(T_{k+1})/D^2)\right) \\
&\leqslant \frac{4T_1GD\ \max\left(1, \log(1+2\log(2T_k)/D^2)\right)}{T} \\
&\leqslant \frac{8GD}{T}\max\left(\rho_T, \frac{8d}{GD\zeta}, \exp\left(\frac{4}{\zeta\lambda}\right)\right)\max\left(1, \log\left(1+\frac{2\log(T)}{D^2}\right)\right).
\end{aligned}$$

This finishes the proof for the case where at least one completed epoch satisfies (4.3.12).

If it is not the case, that means that

$$T \leqslant 4\max(\rho_T, 8d/(GD\zeta), \exp(4/(\zeta\lambda)))$$

And then one can transform the trivial $\mathbb{E}[F(x_1^{k+1})] - F(x^*) \leqslant GD$ into:

$$\mathbb{E}[F(x_1^{k+1})] - F(x^*) \leqslant 4GD\max(\rho_T, 8d/(GD\zeta), \exp(4/(\zeta\lambda)))/T.$$

Putting together the two possible cases about the existence, or not, of a completed epoch satisfying (4.3.12), gives the result for all cases:

$$\mathbb{E}[F(x_1^{k+1})] - F(x^*) \leqslant \frac{8GD}{T}\max\left(\rho_T, \frac{8d}{GD\zeta}, \exp\left(\frac{4}{\zeta\lambda}\right)\right)\max\left(1, \log\left(1+\frac{2\log(T)}{D^2}\right)\right).$$

For T large enough, $\rho_T = \lceil 16\log(-\log_2(\zeta/2) - \log_2(5DG) + 2)/(\zeta GD)\rceil$ so this bound is actually a $O(d\log\log(T)/T)$ as far as d and T are concerned. ∎

Remark: we chose the value of ε_j to guarantee $2\varepsilon_j/(T_j\zeta\lambda) \leqslant 1/4$ in the proof of Lemma 4.14. As the parameter λ (on which depends also ζ) is unknown, we have used $\varepsilon_j = T_j/(2\log(T_j))$, but actually, for any real function ψ that tends to $+\infty$ in $+\infty$, taking $\varepsilon_j = T_j/(\psi(T_j))$ would work, because it would satisfy for j big enough $2\varepsilon_j/(T_j\zeta\lambda) \leqslant 1/4$; and if $\psi \leqslant \log$, it would lead to a better $O(d\log(\psi(T))/T)$ rate.

Moreover, if one knows the strong convexity parameter λ, then it is possible to modify the "Epoch MetaGrad" algorithm so that it leads to $O(d/T)$ bounds (as does the "Epoch Online Newton Step" presented in Section 4.3.2), without the extra $O(\log\log(T))$ factor. It suffices to use $\varepsilon_j = \zeta\lambda T_j/8$ instead of $\varepsilon_j = T_j/(2\log(T_j))$; then one can modify the proof, in particular by using $V'_k = 2^{1-k}GD$ instead of the V_k presented above, to get the result with a $O(d/T)$ bound.

4. Improvements on an online convex optimization algorithm: MetaGrad

Chapter 5

Providing long-term forecast intervals using sequential aggregation

This chapter is a link between the batch and the individual sequences settings. We aim at providing forecast intervals, relying only on external short-term and long-term expert forecasts, without stochastic modeling of the data. However, contrary to the classical individual sequences process, here we do not get any intermediate feedback, and we have to provide at once a whole sequence of forecast intervals for the short and the long term. Thus, we aim at adapting the individual sequences algorithms, which generally require this feedback in their very definition, to this framework.

We introduce a new methodology to do so, which relies on a set of "possible future scenarios", and on an optimization of the algorithms outputs with respect to this set.

We explain how to solve this optimization for three algorithms: exactly for the Ridge regression algorithm, and with some approximations which only widen a bit the resulting intervals for the EWA and Fixed-Share EWA algorithms.

5.1. Introduction

The sequential aggregation provides a robust framework to make day-after-day forecasts, on the basis of the predictions of external experts (see Cesa-Bianchi and Lugosi [2006] for a deep presentation). There are many algorithms available, e.g., the exponentially weighted average forecaster (EWA, first described in Littlestone and Warmuth [1994]), Fixed-Share (Herbster and Warmuth [1998]), or more recently Squint (Koolen and Van Erven [2015]) and ML-Poly (Gaillard et al. [2014]). Some classical linear regression algorithms such as LASSO (introduced in Tibshirani [1996]) or Ridge (Hoerl [1962]) can also be used in this framework.

A short-term objective. The arbitrary sequences setup is very robust, relying on uniform deterministic bounds. Its aim is generally to make single-point (one-step-ahead) forecasts, rather than managing the uncertainty, even if [Gaillard, 2015, Chapter 7], offers some ideas to deal with prediction intervals or distribution functions. The methods in this field rely crucially on the knowledge of the whole set of past observations before the instant of forecast. As a consequence, they are suited for a short term and an online (i.e., "day-after-day") prediction. But they cannot be directly used for a series of long-term forecasts, because of the lack of feedback for the intermediate observations.

Existing methods for long-term forecasts. This issue of long-term forecasts has been tackled in the field of time-series analysis, with methods such as the Box-Jenkins method (cf Box et al. [2015]) or more recent tools (Rinke and Sibbertsen [2016] and Lanne and Saikkonen [2013]). In particular, the issue of managing the uncertainty by forecasting intervals is central in some papers. One can see Chatfield [2001] for a detailed discussion about it, Christoffersen [1998] for an example of a criterion, and Snyder et al. [2001] for a study on an ARIMA model. To compute the probability of simultaneous validity for multiple intervals, Ravishanker et al. [1991] use adapted Bonferroni inequalities.

Nonetheless, the approaches presented in these time-series papers rely on a modeling of the data, rather than on external forecasts of experts.

Aim and outline. In this chapter, we tackle the question of adapting the algorithms of sequential aggregation to long-term forecasts of intervals. In Section 5.2, we provide a framework and a new methodology which benefits from the robustness and the safety of the sequential algorithms. This methodology relies on an optimization upon a large set of "possible sequences of observations". This optimization is a priori computationally costly, but in Sections 5.3 to 5.5, we provide computationally efficient ways of carrying it out for several algorithms. In Section 5.3 we use a closed-form for the Ridge algorithm to compute directly Ridge forecast intervals. In Section 5.4 we tackle the EWA algorithm with a two-step approach: getting "possible sets of weights" from the "possible sequences of observations" (Section 5.4.1) and deriving forecast intervals from these "possible sets of weights" (Section 5.4.2). In Section 5.5, we adapt the approach to get forecast intervals for the Fixed-Share algorithm.

5.2. The forecast intervals framework and methodology

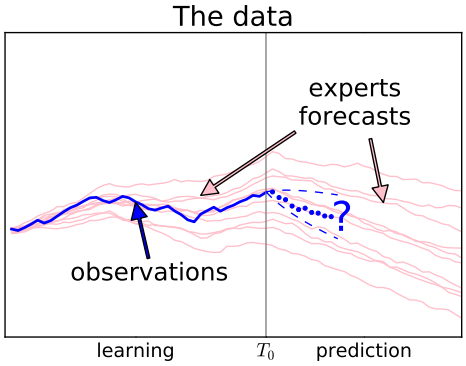


Figure 5.1: Framework of the forecast intervals

We divide time into two periods: learning and prediction.

The first set of rounds $s = 1, \dots, T_0 - 1$ ("learning period") corresponds to the classical setup of online single-point forecasts with expert advice (cf. the beginning of Algorithm 1), see Chapter 2.

The actual goal is to forecast at rounds $t = T_0, T_0+1, \dots, T$ ("prediction period") a quantity y_t (the observation) with the help of K experts, but without any stochastic assumption on the data. One has access to the forecasts of the experts $(f_{k,t})_{1 \leqslant k \leqslant K, 1 \leqslant t < T_0}$ and to the observations $(y_t)_{s<T_0}$ until $T_0 - 1$ (i.e., of the learning period). At each instant t from T_0 to T (the prediction period) the learner has also access to the forecasts $f_{k,t}$ of the experts, and has to predict an interval $[\widehat{y}_t^{\min}, \widehat{y}_t^{\max}]$ aiming at containing y_t. The observation y_t is simultaneously chosen by the environment, but not revealed immediately: the set of observations $(y_t)_{t=T_0,\dots,T}$ is available only after T, i.e., after all the forecast intervals have been made by the learner. In particular, no "feedback" about the quality of the forecasts after T_0 can be used to build the forecast intervals. It is only at the end, after T, that the observations from T_0 to T are revealed and that the accuracy of the forecast intervals can be evaluated.

A new methodology. The sequential aggregation algorithms are well-adapted to tackle online learning without stochastic assumptions, but they can not be directly used here. Indeed, they require the knowledge of the observations up to the previous instant, which are not all available in our framework. The methodology we present here overcomes this issue. It comes from the idea that these algorithms are robust and able to deal with all sets of observations, even worst-case ones. Therefore, forecast intervals that contain all the possible future forecasts of the algorithm are likely to contain the observation. That is exactly how we build them.

Methodology (formalized in **Algorithm 15**):

Algorithm 14 The framework of forecast intervals

I. Learning period
for $s = 1, \dots, T_0 - 1$:

1. The observation y_s is chosen by the environment
2. Get the expert forecasts $(f_{k,s})_{1 \leqslant k \leqslant K}$
3. Provide a single-point forecast $\widehat{y}_s = \sum_{k=1}^{K} u_{k,s} f_{k,s}$
4. Observe y_s

II. Prediction period
for $t = T_0, T_0 + 1, \dots, T$:

1. The observation y_t is chosen by the environment but not revealed
2. Get $(f_{k,t})_{1 \leqslant k \leqslant K}$
3. Provide a forecast interval $[\widehat{y}_t^{\min}, \widehat{y}_t^{\max}]$

for $t = T$:
Observe $y_{T_0}, \dots, y_T$

The starting point is to assume that the sequence of observations to be predicted $(y_{T_0}, \dots, y_T)$ can be any element of a large subset $S \subset \mathbb{R}^{T-T_0+1}$, the set of the "possible scenarios for the future". At round t, for a given aggregation algorithm, the $t-T_0$ first components $(z_{T_0}, \dots, z_{t-1})$ of any "possible scenario" $(z_{T_0}, \dots, z_T) \in S$, in combination with the expert forecasts, lead to a prediction $\widehat{z}_t$: the prediction that would be made at round t in the classical sequential setting, if the observations sequentially revealed had been $y_1, \dots, y_{T_0-1}, z_{T_0}, \dots, z_{t-1}$. So the set S of possible scenarios leads to a set of "possible predictions at round t", which we denote $\widehat{S}_t := \big\{\widehat{z}_t : (z_{T_0}, \dots, z_{t-1}) \text{ is the prefix of a sequence in } S\big\}$. The forecast interval at round t is then defined as the smallest interval containing $\widehat{S}_t$.

The set S of the "possible scenarios". The choice of this subset S is important. It is the main "modeling" choice of the statistician, most of the "data modeling" being supposedly included into the expert forecasts. It can be for example a product of intervals: $S = \prod_{t=T_0}^{T} [\underline{B}_t, \overline{B}_t]$. The set S is typically constructed based on y_{T_0} and on what we know about the evolution of the observations. E.g., if one assumes that the absolute variation of the observations between two time steps is bounded by some value Δ, then the observation y_t will lie between:

$$\underline{B}_t = y_{T_0} - (t - T_0)\Delta \qquad \text{and} \qquad \overline{B}_t = y_{T_0} + (t - T_0)\Delta \qquad \text{for} \quad t \geqslant T_0.$$

The set S is in this case a cone, that will contain all the observations. It is this approach that is shown in Figure 5.2 where the red cone contains all the considered sequences of observations (three of them are drawn in green).

Algorithm 15 Methodology

Preliminaries:
Observe $(y_0, \dots, y_{T_0-1})$

for $t = T_0, \dots, T$:
I. Building $\widehat{S}_t$:
Initialize $\widehat{S}_t = \emptyset$
for each $(z_{T_0}, \dots, z_T) \in S$:
1. Feed any classical learning algorithm with $(y_0, \dots, y_{T_0-1}, z_{T_0}, \dots, z_{t-1})$ and $(f_{k,\tau})_{1 \leqslant k \leqslant K, 1 \leqslant \tau \leqslant t}$
2. Predict $\widehat{z}_t$
3. Update $\widehat{S}_t \leftarrow \widehat{S}_t \cup \{\widehat{z}_t\}$

II. Output:
Output the forecast interval $[\widehat{y}_t^{\min}, \widehat{y}_t^{\max}]$ defined as the smallest interval containing $\widehat{S}_t$

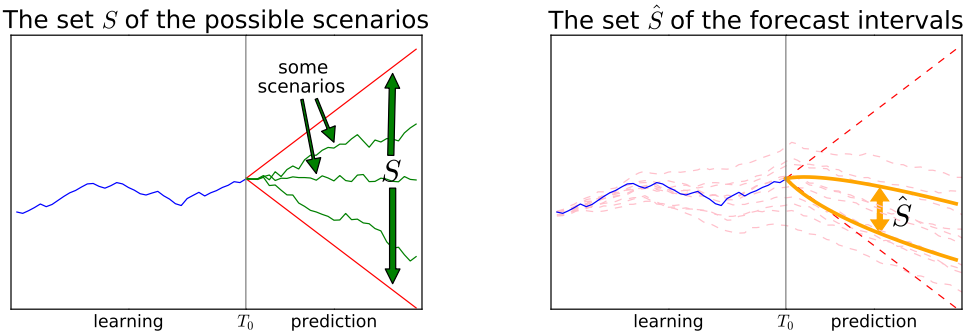


Figure 5.2: The set S of the possible scenarios (left, green), combined with the expert forecasts (right, pink), leads to the set $\widehat{S} = (\widehat{S}_t)_{t \geqslant T_0}$ of forecast intervals (right, orange).

Computational issue to solve

A difficulty of this approach is that, as soon as the subset S is large (infinite, possibly uncountable), one just cannot compute the forecasts scenario by scenario (S contains infinitely many of them!). But we actually aim only at the higher and lower forecasts $\widehat{y}_t^{\max}$ and $\widehat{y}_t^{\min}$, which is an optimization problem over the set of the possible scenarios up to round t. This optimization depends heavily on the algorithm, and each algorithm requires a different approach.

We give in Section 5.3 an example of an algorithm (Ridge) for which a (linear in some sense) closed-form expression allows to compute efficiently and quickly the intervals. We then present in Section 5.4 a way to deal with the EWA algorithm, using a split of the weight computations and a separation from the past (resulting in a slightly wider range of possible weights). We then adapt this approach in Section 5.5 for the Fixed-Share algorithm.

5.3. Forecast intervals with the Ridge regression forecaster

In this section, we address the Ridge algorithm, for which a closed-form expression of the forecasts (Subsection 5.3.1) allows to compute directly the forecast intervals (Subsection 5.3.2).

5.3.1. The Ridge regression forecaster

The Ridge regression forecaster is a regularized "least squares" estimator, introduced by Hoerl [1962]. It relies on a regularization parameter λ, and on a Euclidean regularization $u \mapsto \|u\|_2^2$.

Consider some instant t, and denote:

- by f_t the vector of all the experts' forecasts at time t: $f_t = (f_{1,t}, \dots, f_{K,t})^\top$,
- by $F_t = (f_{s,k})_{\substack{s \leqslant t \\ k \leqslant K}}$ the $t \times K$ matrix of the experts' forecasts up to time t (notice that we inverse the order of indices of the experts' forecasts),
- by $Y_t = (y_1, \dots, y_t)^\top$ the column-vector of the observations up to time t ($.^\top$ meaning "transpose").

The weight vector output by Ridge $u_{t+1}^R := \begin{pmatrix} u_{1,t+1}^R \\ \dots \\ u_{K,t+1}^R \end{pmatrix}$ is the minimizer of:

$$u = \begin{pmatrix} u_{1,t+1} \\ \dots \\ u_{K,t+1} \end{pmatrix} \longmapsto \|Y_t - F_t u\|_2^2 + \lambda \|u\|_2^2. \tag{5.3.1}$$

We recall now Lemma 2.6 (proved in Chapter 2), which shows that the weight vector generated by Ridge, and therefore the Ridge regression forecasts, are linear mappings of the observations.

Lemma 5.1. *The weights and forecasts provided by the Ridge algorithm (with regularization parameter λ) are linear with respect to the observations vector Y_t:*

$$\begin{pmatrix} u_{1,t+1}^R \\ \dots \\ u_{K,t+1}^R \end{pmatrix} = M_t Y_t \qquad \text{with } M_t := (F_t^\top F_t + \lambda I_K)^{-1} F_t^\top$$

thus

$$\widehat{y}_{t+1} = V^{t+1} Y_t \qquad \text{with } V^{t+1} := (f_{1,t+1}, \dots, f_{K,t+1})(F_t^\top F_t + \lambda I_K)^{-1} F_t^\top \in \mathbb{R}^t$$

5.3.2. The forecast intervals for the Ridge regression

Recall that the observations y_s are known for all instants $s < T_0$ of the training period. We consider the following known bounds:

$$\underline{B}_s \leqslant y_s \leqslant \overline{B}_s \text{ for any instant } T_0 \leqslant s \leqslant t. \tag{H_t}$$

Lemma 5.2. *The forecast intervals* $[\widehat{y}_t^{\min}, \widehat{y}_t^{\max}]$ *output by Ridge following the methodology given in Section 5.2 and under the bounds:* $\underline{B}_s \leqslant y_s \leqslant \overline{B}_s$ *for any instant* $T_0 \leqslant s \leqslant t$*, satisfy:*

$$\widehat{y}_t^{\max} = \sum_{s=0}^{T_0-1} V_s^{t+1} y_s + \sum_{s=T_0}^{t} V_s^{t+1} (\overline{B}_s \mathbf{1}_{V_s^{t+1} \geqslant 0} + \underline{B}_s \mathbf{1}_{V_s^{t+1} < 0})$$

and

$$\widehat{y}_t^{\min} = \sum_{s=0}^{T_0-1} V_s^{t+1} y_s + \sum_{s=T_0}^{t} V_s^{t+1} (\underline{B}_s \mathbf{1}_{V_s^{t+1} \geqslant 0} + \overline{B}_s \mathbf{1}_{V_s^{t+1} < 0}).$$

Proof. Writing $\widehat{y}_{t+1} = \sum_{u=1}^{t} V_u^{t+1} y_u$ emphasizes that the highest possible value for $\widehat{y}_{t+1}$ corresponds to a scenario in which the value of the observation at any time u (such that $T_0 \leqslant u \leqslant t$) is minimal if $V_u^{t+1} < 0$ and maximal if $V_u^{t+1} \geqslant 0$.

On the contrary, $\widehat{y}_{t+1}$ reaches its lowest possible value when the value of the observation at any time u (such that $T_0 \leqslant u \leqslant t$) is maximal if $V_u^{t+1} < 0$ and minimal if $V_u^{t+1} \geqslant 0$. ∎

Some remarks on the Ridge forecast intervals. The width of the forecast interval at time t is a sum of t non-negative terms (which change at each t). It scales linearly with the set of bounds $\{\underline{B}_u, \overline{B}_u\}_{u \leqslant t}$, in particular, it does not necessarily belong to the interval generated by the expert forecasts.

5.4. Forecast intervals with the EWA algorithm

In this section, we address the exponentially weighted average (EWA) algorithm. Contrary to the Ridge algorithm, it does not provide an easy (e.g., linear) link between the observations and the forecasts. So optimizing directly upon the whole set of possible observations scenarios (e.g., $\prod_{t=T_0}^{T} [\underline{B}_t, \overline{B}_t]$) seems to be computationally intractable. In this section, we therefore present a compromise: using computationally cheap methods, at the cost of being less sharp and having slightly larger weight intervals.

Our approach. We adopt a two-step approach. First, for any round t, on the basis of the set S of the possible scenarios and on the expert forecasts, we compute separately for each expert an interval of possible weights $[p_{j,t}^{\min}, p_{j,t}^{\max}]$. We provide in Subsection 5.4.1 a computationally efficient way to do so. The second step consists in passing from weight intervals to forecast intervals. This amounts to choosing a weight vector compatible with each weight interval, and see which highest and lowest possible forecasts can be obtained this way. We implicitly accept thus to aggregate weights that do not come from the same observations scenario. We provide in Subsection 5.4.2 fast and easy ways to deal with two aggregation problems in this setup: linear and convex aggregations. It is the convex aggregation case that corresponds to the EWA algorithms; the linear case is presented to allow future use of linear algorithms.

5.4.1. Weight intervals for the EWA (with fixed learning rate) algorithm

Our trick. The method we present here consists in splitting the computations of the weights for each expert, and obtaining them by successive updates.

We provide weight intervals updates that only take into account the next forecasts, the weight to be updated and bounds on observations (but no past data, and no other weight). That is, we restrict our attention to weight updates for which, for any j and t, the next weight $p_{j,t+1}$ must only be a function of $p_{j,t}$, of $f_{t+1} = (f_{1,t+1}, \dots, f_{K,t+1})$ and of known bounds on the observations.

This scheme leads to a significant decrease in the computation time, as there are only around K variables taken into account. It leads to a slight broadening of the weight intervals.

We focus here on the EWA algorithm, because its definition relies on a multiplicative update of the weights, so it is well adapted to our approach. Recall that in the EWA algorithm with fixed learning rate η, the weights are convex and satisfy:

$$p_{j,t+1} = \frac{p_{j,t}}{\sum_i p_{i,t}\alpha_{i,j,t}}$$

where $\alpha_{i,j,t} = \exp\Big(-\eta\big(y_t - f_{i,t}\big)^2 + \eta\big(y_t - f_{j,t}\big)^2\Big) = \exp\Big(\eta\big(f_{j,t}^2 - f_{i,t}^2 + 2y_t(f_{i,t} - f_{j,t})\big)\Big)$.

This quantity $\alpha_{i,j,t}$, which contains the experts interactions in the evolution of the weights of EWA, will be key in our arguments.

Warm-up: straightforward but loose bound.
The quantity $\max\big\{(y_t - f_{j,t})^2 : t \in \{T_0, .., T\} \text{ and } j \in \{1, .., K\}\big\}$ is unknown beforehand, but if one has access to an upper bound Q on it, then one gets directly the following bounds on the weights (proved in the supplementary material).

Lemma 5.3. *For any j, the weights of the j-th expert satisfy:*

$$p_{j,t}^{\min} = p_{j,T_0}\exp\big(-(t - T_0)\eta Q\big) \leqslant p_{j,t} \leqslant 1 - (1 - p_{j,T_0})\exp\big(-(t - T_0)\eta Q\big) = p_{j,t}^{\max}.$$

This bound is not sharp, in particular it does not take into account the real positions of the experts. Also, if $t\eta Q$ is much larger than one (which happens in many practical applications), then the lower bound in the lemma is nearly null and the upper bound is close to 1. This leads to a lowest and a highest forecast that correspond respectively to the highest and the lowest forecast of the experts. Such an interval, too wide and relying only on two extreme experts, is not very relevant in practice.

So we show hereunder how to use more accurately the information at our disposal, to get sharper bounds on the weights.

More accurate bounds. So far, we have not used any precise information on the set S of the possible observations, and we will do so now. We make the same assumption (H_t) as for Ridge: we assume that at each instant τ, there are known bounds $\underline{B}_\tau$ and $\overline{B}_\tau$ for the

observation: $\underline{B}_\tau \leqslant y_\tau \leqslant \overline{B}_\tau$. For given j and t, define:

$$\begin{aligned} M_{j,t} &= \max_{y_t \in [\underline{B}_t, \overline{B}_t]} \max_{i \in \{1\dots K\}} \alpha_{i,j,t} \\ &= \max_{y_t \in [\underline{B}_t, \overline{B}_t]} \max_{i \in \{1\dots K\}} \exp\Big(-\eta\big(y_t - f_{i,t}\big)^2 + \eta\big(y_t - f_{j,t}\big)^2\Big) \\ \text{and} \qquad m_{j,t} &= \min_{y_t \in [\underline{B}_t, \overline{B}_t]} \min_{i \in \{1\dots K\}} \alpha_{i,j,t} \\ &= \min_{y_t \in [\underline{B}_t, \overline{B}_t]} \min_{i \in \{1\dots K\}} \exp\Big(-\eta\big(y_t - f_{i,t}\big)^2 + \eta\big(y_t - f_{j,t}\big)^2\Big). \end{aligned}$$

These two quantities will be studied in deeper details, with a more explicit computation, in Lemma 5.5. For the moment we only use their definition.

The following forecast interval $[p_{j,t}^{\min}, p_{j,t}^{\max}]$ only makes the weight intervals larger:

$$p_{j,t}^{\min} = \frac{p_{j,T_0}^{\min}}{p_{j,T_0}^{\min} + (1 - p_{j,T_0}^{\min}) \prod_{u=T_0}^{t-1} M_{j,u}} \qquad \text{and} \qquad p_{j,t}^{\max} = \frac{p_{j,T_0}^{\max}}{p_{j,T_0}^{\max} + (1 - p_{j,T_0}^{\max}) \prod_{u=T_0}^{t-1} m_{j,u}} \tag{5.4.1}$$

Contrary to the bounds in Lemma 5.3, here the real forecast of the expert is taken into account via $M_{j,u}$ and $m_{j,u}$, which prevents one from an overcautious uniform evolution of the weights.

Proof for these bounds. The facts that $\alpha_{j,j,t} = 1$ and that $\sum_i p_{i,t} = 1$ lead to:

$$p_{j,t+1} \geqslant \frac{p_{j,t}}{p_{j,t} + \sum_{i \neq j} p_{i,t} M_{j,t}} = \frac{p_{j,t}}{p_{j,t} + (1 - p_{j,t}) M_{j,t}} \quad \text{and} \quad p_{j,t+1} \leqslant \frac{p_{j,t}}{p_{j,t} + (1 - p_{j,t}) m_{j,t}}.$$

Thus, a reasonable update for the weight intervals, that can make them (and the forecast intervals) only larger, is:

$$p_{j,t+1}^{\min} = \frac{p_{j,t}^{\min}}{p_{j,t}^{\min} + (1 - p_{j,t}^{\min}) M_{j,t}} \qquad \text{and} \qquad p_{j,t+1}^{\max} = \frac{p_{j,t}^{\max}}{p_{j,t}^{\max} + (1 - p_{j,t}^{\max}) m_{j,t}}.$$

One can then deduce (by an induction given in the supplementary material) the direct complete form of the weight intervals.

Lemma 5.4. *The previous updates lead to the weight intervals* $[p_{j,t}^{\min}, p_{j,t}^{\max}]$ *given in* (5.4.1).

Final piece: efficient computation. It remains to be able to compute efficiently $M_{j,t}$ and $m_{j,t}$, by identifying which observation and which expert forecast maximizes or minimizes $\alpha_{i,j,t}$. That is what the next lemma does.

Lemma 5.5. *For given j and t, the maximum $M_{j,t}$ of $\alpha_{i,j,t}$ is obtained for an observation equal to one of the bounds $\underline{B}_t$ or $\overline{B}_t$, and for the expert that is the closest to this bound:*

$$\operatorname*{argmax}_{(y_t,i)} \alpha_{i,j,t} \cap \left\{ \left(\underline{B}_t,\ \operatorname*{argmin}_{i\in\{1,\dots,K\}} |f_{i,t} - \underline{B}_t| \right), \left(\overline{B}_t,\ \operatorname*{argmin}_{i\in\{1,\dots,K\}} |f_{i,t} - \overline{B}_t| \right) \right\} \neq \emptyset$$

As for the minimum $m_{j,t}$ of $\alpha_{i,j,t}$, it is obtained for an observation equal to one of the bounds $\underline{B}_t$ or $\overline{B}_t$, and for the expert that is the farthest to this bound:

$$\operatorname*{argmin}_{(y_t,i)} \alpha_{i,j,t} \cap \left\{ \left(\underline{B}_t,\ \operatorname*{argmax}_{i\in\{1,\dots,K\}} |f_{i,t} - \underline{B}_t| \right), \left(\overline{B}_t,\ \operatorname*{argmax}_{i\in\{1,\dots,K\}} |f_{i,t} - \overline{B}_t| \right) \right\} \neq \emptyset.$$

The proof, detailed in the supplementary material, relies on the study (for a given instant t and an expert j) of the function $(y_t, f_{i,t}) \mapsto \alpha_{i,j,t}$.

High-level remarks. An important remark is that, since we allow in this section to separate the weight intervals computations, we get a wider range of weights than what a direct optimization upon the possible scenarios would have provided, and so this leads to larger forecast intervals.

As EWA is a convex algorithm, passing from weight intervals to forecast intervals requires normalization (cf. Subsection 5.4.2), so the forecast interval is always inside the expert forecasts interval. As a consequence, the choice of the possible scenarios, though still important, is less crucial than for non-convex algorithms such as Ridge.

5.4.2. From weight intervals to forecast intervals

Section 5.4.1 shows how to get weight intervals. It then remains to form weight vectors from these intervals. Two aggregation frameworks are studied hereunder: linear aggregation and convex aggregation.

A preliminary example: linear aggregation. The case of linear aggregation (i.e., with weights in $\mathbb{R}$ and without normalization), is straightforward. The maximum M and the minimum m of $\left\{ \sum_{j=1}^{K} u_j f_j :\ u_1 \in [u_1^{\min}, u_1^{\max}], \dots,\ u_K \in [u_K^{\min}, u_K^{\max}] \right\}$ are given as follows:

Lemma 5.6.

$$M = \sum_{j=1}^{K} \left(u_j^{\min} f_j \mathbf{1}_{f_j<0} + u_j^{\max} f_j \mathbf{1}_{f_j\geqslant 0} \right) \qquad \text{and} \qquad m = \sum_{j=1}^{K} \left(u_j^{\max} f_j \mathbf{1}_{f_j<0} + u_j^{\min} f_j \mathbf{1}_{f_j\geqslant 0} \right).$$

Proof. If $f_j > 0$, then for all $u \in [u_j^{\min}, u_j^{\max}]$, one has $u_j^{\min} f_j \leqslant u f_j \leqslant u_j^{\max} f_j$; and if $f_j < 0$, then for all $u \in [u_j^{\min}, u_j^{\max}]$, one has $u_j^{\max} f_j \leqslant u f_j \leqslant u_j^{\min} f_j$. ∎

Convex aggregation

In this case, one is given $f = (f_1 \dots, f_K) \in \mathbb{R}^K$ and $[p_1^{\min}, p_1^{\max}], \dots, [p_K^{\min}, p_K^{\max}] \subset [0,1]$, but this time the weights have to be normalized. One aims at computing:

$$\max \left\{ \sum_{j=1}^{K} \frac{p_j}{\sum_k p_k} f_j : \ p_1 \in [p_1^{\min}, p_1^{\max}], \dots, \ p_K \in [p_K^{\min}, p_K^{\max}] \right\} \qquad (*)$$

$$\min \left\{ \sum_{j=1}^{K} \frac{p_j}{\sum_k p_k} f_j : \ p_1 \in [p_1^{\min}, p_1^{\max}], \dots, \ p_K \in [p_K^{\min}, p_K^{\max}] \right\} \qquad (**)$$

and the corresponding argmax (∗) and argmin (∗∗).

We assume that at least one weight $p_j^{\min}$ is positive, so all expressions are well-defined. We focus on the first problem (∗), the second one (∗∗) is addressed by symmetry below.

Discretization of the problem. The next lemma shows that what seems a continuous optimization problem is actually a discrete optimization problem.

Lemma 5.7. *The argmax (∗) contains at least one element for which all the weights are either maximal or minimal:*

$$\left(\operatorname{argmax} \left\{ \sum_{j=1}^{K} \frac{p_j}{\sum_k p_k} f_j \right\} \right) \cap \left(\{p_1^{\min}, p_1^{\max}\} \times \dots \times \{p_K^{\min}, p_K^{\max}\} \right) \neq \emptyset.$$

The proof, given in the supplementary material relies on the idea that if a weight p_j is different from $p_j^{\min}$ and $p_j^{\max}$, then (in most cases) either increasing or decreasing p_j will increase $\sum_{j=1}^{K} p_j f_j / \sum_k p_k$.

Efficient computation via a ranking. The previous lemma shows that there exists a weight vector leading to a maximal forecast, and whose components are (before normalization) $p_j^{\min}$ or $p_j^{\max}$. So one only has to test the 2^K vectors of $\left(\{p_1^{\min}, p_1^{\max}\} \times \dots \times \{p_K^{\min}, p_K^{\max}\} \right)$ and get the one which leads to the highest value. But 2^K is quite a lot, and actually, one can show that a "ranking" of the coordinates allows one to test only $K+1$ weight vectors. These $K+1$ vectors are the ones that "give their maximum possible weights to the experts that forecast the highest values and the minimum possible weight to the others", which is quite intuitive.

Lemma 5.8. *Denote by $n_1, \dots, n_K$ the indexes of the coordinates of f sorted by decreasing order. There exists $R \leqslant K$ and a weight vector v in the argmax of Lemma 5.7 such that:*

$$v_{n_1} = p_{n_1}^{\max}, \dots, v_{n_R} = p_{n_R}^{\max} \qquad \text{and} \qquad v_{n_{R+1}} = p_{n_{R+1}}^{\min}, \dots, v_{n_K} = p_{n_K}^{\min}.$$

The proof, based on similar ideas as the previous one, can be found in the supplementary material.

From maximization to minimization. Let us move now to the second problem: the minimum (and argmin) problem. It can be solved using the argmax result, with a symmetry argument: it suffices to replace f_j by $-f_j$.

$$\operatorname{argmin}\left\{\sum_{j=1}^{K}\frac{p_j}{\sum_k p_k}f_j:\ p_1\in[p_1^{\min},p_1^{\max}],\dots,\ p_K\in[p_K^{\min},p_K^{\max}]\right\}=$$
$$\operatorname{argmax}\left\{\sum_{j=1}^{K}\frac{p_j}{\sum_k p_k}(-f_j):\ p_1\in[p_1^{\min},p_1^{\max}],\dots,\ p_K\in[p_K^{\min},p_K^{\max}]\right\}$$

A decreasing order on $(-f_1,\dots,-f_K)$ corresponds to an increasing order on $(f_1,\dots,f_K)$. This directly leads to the following lemma (the difference with the previous one is that an increasing order is used, instead of a decreasing one).

Lemma 5.9. *Denote by $n_1,\dots,n_K$ the indexes of the coordinates of f sorted by increasing order. There exists $R\leqslant K$ and a weight vector v in the argmin such that:*

$$v_{n_1}=p_{n_1}^{\max},\dots,v_{n_R}=p_{n_R}^{\max}\qquad\text{and}\qquad v_{n_{R+1}}=p_{n_{R+1}}^{\min},\dots,v_{n_K}=p_{n_K}^{\min}.$$

Similarly to the maximization problem, only $K+1$ vectors need to be tested.

Remark. One can note that even in the cases where the weight intervals get wider over time, it is not sufficient to guarantee that the forecast intervals will also get wider, since they depend on the expert forecasts. Even if the set of expert forecasts get broader over time (i.e., the extremal expert forecasts are moving away from each other), it is not enough to guarantee that the size of the forecast intervals will increase, since it is mostly driven by the experts with the highest weights, which are not necessarily the extremal experts. To sum up, there is no standard evolution of the forecast intervals, it is really the behaviour of the experts (along with the chosen algorithm and the possible scenarios) that will determine the forecast intervals.

5.5. Extension to the Fixed-Share algorithm

The Fixed-Share algorithm. The approach developed hereabove can be applied to another important algorithm, derived from EWA: Fixed-Share EWA. This algorithm involves two parameters: $\alpha\in[0,1]$ and $\eta>0$, and its weights satisfy:

$$p_{j,t+1}=(1-\alpha)\frac{p_{j,t}}{\sum_i p_{i,t}\alpha_{i,j,t}}+\frac{\alpha}{K}$$

where, as in EWA,

$$\alpha_{i,j,t}=\exp\Big(-\eta(y_t-f_{i,t})^2+\eta(y_t-f_{j,t})^2\Big).$$

Fixed-Share forecast intervals. Minimizing (or maximizing) $p_{j,t+1}$ is equivalent to minimizing (or maximizing) $p_{j,t}/\left(\sum_i p_{i,t}\alpha_{i,j,t}\right)$, so the same approach as for EWA can be applied.

Define again $M_{j,t} = \max_{(y_t,i)} \alpha_{i,j,t}$, and $m_{j,t} = \min_{(y_t,i)} \alpha_{i,j,t}$. They can be computed by Lemma 5.5. One has then upper and lower bound on the weights:

$$p_{j,t+1} \geqslant \alpha \frac{p_{j,t}}{p_{j,t} + (1-p_{j,t})M_{j,t}} + \frac{\alpha}{K} \qquad \text{and} \qquad p_{j,t+1} \leqslant \alpha \frac{p_{j,t}}{p_{j,t} + (1-p_{j,t})m_{j,t}} + \frac{\alpha}{K}.$$

Theses inequalities provide reasonable updates (which, as in the EWA case, will make the weight and forecast intervals only larger):

$$p^{\min}_{j,t+1} = \alpha \frac{p^{\min}_{j,t}}{p^{\min}_{j,t} + (1-p^{\min}_{j,t})M_{j,t}} + \frac{\alpha}{K} \qquad \text{and} \qquad p^{\max}_{j,t+1} = \alpha \frac{p^{\max}_{j,t}}{p^{\max}_{j,t} + (1-p^{\max}_{j,t})m_{j,t}} + \frac{\alpha}{K}.$$

Then, one can apply the methodology of Subsection 5.4.2 to get forecast intervals from the weight intervals.

5.6. Lines of future works

The methods in this chapter for the computation of the forecast intervals rely heavily on known closed-forms for the algorithms weights. It would be interesting to try to apply the methodology on other algorithms, especially some for which no closed-form formula is known, e.g., the LASSO algorithm.

A line of thought would be to try and get some theoretical guarantees. This requires to define new and adapted benchmarks and criteria of performance. Indeed, the lack of modeling on the observations process prevents any guarantee on the proportion of observations falling inside the forecast intervals (unless of course the forecast intervals contain the totality of the possible observations, which is a trivial case). And many other criteria such as the quadratic loss do not make obvious sense in the forecast intervals setup.

5.7. Supplementary material

Proof of Lemma 5.3
For any i, j, t, the coefficient $\alpha_{i,j,t}$ satisfy $0 < \alpha_{i,j,t} \leqslant \exp(\eta Q)$. So for any j, one has:

$$p_{j,t+1} \geqslant \frac{p_{j,t}}{\sum_i p_{i,t} \exp(\eta Q)} = \exp(-\eta Q) \frac{p_{j,t}}{\sum_i p_{i,t}} = p_{j,t} \exp(-\eta Q)$$

(recall that $\sum_i p_{i,t} = 1$). By an immediate induction, $p_{j,t} \geqslant p_{j,0} \exp(-t\eta Q)$.

As a consequence, applying this bound to all the other experts, and using twice the fact that the weights sum up to 1, leads to:

$$p_{j,t} = 1 - \sum_{i \neq j} p_{i,t} \leqslant 1 - \sum_{i \neq j} p_{i,0} \exp(-t\eta Q) = 1 - (1 - p_{j,0}) \exp(-t\eta Q). \qquad \square$$

Proof of Lemma 5.4
$p^{\min}_{j,t+1} = \dfrac{p^{\min}_{j,t}}{p^{\min}_{j,t} + (1 - p^{\min}_{j,t}) M_{j,t}}$ can be re-written as: $\dfrac{1}{p^{\min}_{j,t+1}} - 1 = M_{j,t} \left(\dfrac{1}{p^{\min}_{j,t}} - 1 \right)$.

A direct induction (starting at T_0) gives:

$$\frac{1}{p^{\min}_{j,t}} - 1 = \left(\frac{1}{p^{\min}_{j,T_0}} - 1 \right) \prod_{u=T_0}^{t-1} M_{j,u}$$

$$\text{so that} \qquad p^{\min}_{j,t} = \frac{p^{\min}_{j,T_0}}{p^{\min}_{j,T_0} + (1 - p^{\min}_{j,T_0}) \prod_{u=T_0}^{t-1} M_{j,u}}.$$

The same computation (replacing $M_{j,t}$ by $m_{j,t}$) holds for $p^{\max}_{j,t}$ and leads to the desired result. $\square$

Proof of Lemma 5.5
Let us consider the expert j at time t (whose prediction is $f_{j,t}$), and tackle the maximization and minimization of $\alpha_{i,j,t}$ with respect to y_t and to $f_{i,t}$ (with $f_{i,t} \in \{f_{j,t}\}_{j=1..K}$).

Split the experts into three groups: the ones that forecast higher than $f_{j,t}$ (group 1), the ones that forecast lower than $f_{j,t}$ (group 2), and the ones that forecast $f_{j,t}$ (group 3).

One has: $\dfrac{\partial \alpha_{i,j,t}}{\partial y_t} = 2(f_{i,t} - f_{j,t}) \alpha_{i,j,t}$ and $\alpha_{i,j,t} > 0$; so $\dfrac{\partial \alpha_{i,j,t}}{\partial y_t} > 0 \Leftrightarrow (f_{i,t} - f_{j,t}) > 0$.

For the experts of the group 1, $\alpha_{i,j,t}$ is increasing with y_t. So its maximum is attained for the maximal value of y_t: $y_t = \overline{B}_t$; and its minimum is attained for the minimum value of y_t: $y_t = \underline{B}_t$. On the contrary, for the experts of the group 2, $\alpha_{i,j,t}$ is maximal for $y_t = \underline{B}_t$ and minimal for $y_t = \overline{B}_t$. As for the experts of the group 3, the value of y_t has no influence, for them $\alpha_{i,j,t} = 1$ whatever y_t is.

As a consequence, there are only two values of y_t to consider in order to maximize or minimize (in y_t) $\alpha_{i,j,t}$: $\underline{B}_t$ and $\overline{B}_t$. Hence:

$$\max_{(y_t,i)} \alpha_{i,j,t} = \max_{y_t \in \{\underline{B}_t, \overline{B}_t\}} \max_{i \in \{1\dots K\}} \alpha_{i,j,t} = \max_{B \in \{\underline{B}_t, \overline{B}_t\}} \left\{ \max_{i \in \{1\dots K\}} \overline{\alpha_{i,t}}(B) \right\} \tag{5.7.1}$$

where $\overline{\alpha_{i,t}}(B) = \exp\left(-\eta\,(B - f_{i,t})^2 + \eta\,(B - f_{j,t})^2\right)$ is decreasing in $(B - f_{i,t})^2$ and therefore in $|B - f_{i,t}|$. As a consequence, $\operatorname*{argmax}_{i \in \{1\dots K\}} \overline{\alpha_{i,t}}(\underline{B}_t) = \operatorname*{argmin}_{i \in \{1\dots K\}} |\underline{B}_t - f_{i,t}|$ and $\operatorname*{argmax}_{i \in \{1\dots K\}} \overline{\alpha_{i,t}}(\overline{B}_t) = \operatorname*{argmin}_{i \in \{1\dots K\}} |\overline{B}_t - f_{i,t}|$.

Therefore, Equation (5.7.1) shows that either the pair $\left(\underline{B}_t, \operatorname*{argmin}_i |f_{i,t} - \underline{B}_t|\right)$ or the pair $\left(\overline{B}_t, \operatorname*{argmin}_i |f_{i,t} - \overline{B}_t|\right)$ belong to $\operatorname*{argmax}_{(y_t,i)} \alpha_{i,j,t}$.

For $\operatorname*{argmin}_{(y_t,i)} \alpha_{i,j,t}$, we proceed in a symmetric way, since $\min_{(y_t,i)} \alpha_{i,j,t} = \min_{B \in \{\underline{B}_t, \overline{B}_t\}} \left\{ \min_{i \in \{1\dots K\}} \overline{\alpha_{i,t}}(B) \right\}$. □

Proof of Lemma 5.7

Let f be the vector of the expert forecasts: $f = (f_1, \dots, f_K)$, and recall that the set of the possible weights before normalization is $\prod_j [p_j^{\min}, p_j^{\max}]$, with all p_j being non-negative, and at least one of them being positive.

Denote by W the set of these possible weights after normalization:

$$W = \left\{ \left(\frac{p_1}{\sum_k p_k}, \dots, \frac{p_K}{\sum_k p_k} \right) :\ p_1 \in [p_1^{\min}, p_1^{\max}], \dots,\ p_K \in [p_K^{\min}, p_K^{\max}] \right\}.$$

It is clear that W is a compact set (because it is the image of a compact set by the mapping $x \mapsto (1/\|x\|_1)x$ which is continuous outside the null vector, thus on the set at hand). Problem $(*)$ is equivalent to getting the maximum of the function $w \in W \mapsto w \cdot f$, which is continuous (because it is linear in finite dimension). Therefore it reaches its maximum on the compact set W, so the argmax $(*)$ is not empty.

We will show that the argmax $(*)$ contains at least one element located (before normalization) at a corner of $\prod_j [p_j^{\min}, p_j^{\max}]$. To do so, let u be a weight vector belonging to the argmax $(*)$, before normalization, and let u' be obtained by replacing the j-th component u_j of u by another value u'_j. Denote by S_u the "normalized dot product": $S_u = u \cdot f / \|u\|_1$. Then:

$$\begin{aligned}
\frac{u' \cdot f}{\|u'\|_1} &= \frac{(u'_j - u_j) f_j}{\|u'\|_1} + \sum_{i=1}^{K} \frac{u_i f_i}{\|u'\|_1} \\
&= \frac{(u'_j - u_j)(f_j - S_u)}{\|u'\|_1} + \frac{(u'_j - u_j) S_u}{\|u'\|_1} + \sum_{i=1}^{K} \frac{u_i f_i}{\|u'\|_1} \\
&= \frac{(u'_j - u_j)(f_j - S_u)}{\|u'\|_1} + \frac{(\|u'\|_1 - \|u\|_1) S_u}{\|u'\|_1} + \frac{\|u\|_1 S_u}{\|u'\|_1} \\
&= S_u + \frac{(u'_j - u_j)(f_j - S_u)}{\|u'\|_1}.
\end{aligned} \tag{5.7.2}$$

The first equality comes from the fact that u and u' only differ on their j-th component. The second one introduces S_u by writing $f_j = \big(f_j - S_u\big) + S_u$. The third one is based on $\|u'\|_1 - \|u\|_1 = u'_j - u_j$ and on $S_u \|u\|_1 = u \cdot f$.

Three cases appear for the coordinates of u.

First case: $f_j = S_u$. Then the second term of the sum (5.7.2) is null, the value of u'_j has no impact and can be replaced at will by $p_j^{\min}$ or $p_j^{\max}$.

Second case: $f_j > S_u$. If $u_j \neq p_j^{\max}$, consider $u'_j > u_j$, then the second term of the sum (5.7.2) is positive, and one has $S_u < u' \cdot f / \|u'\|_1$; this is not possible since u belongs to the argmax. So $u_j = p_j^{\max}$.

Third case: $f_j < S_u$. If $u_j \neq p_j^{\min}$, consider $u'_j < u_j$, then the second term of the sum (5.7.2) is positive, and one has $S_u < u' \cdot f / \|u'\|_1$; this is not possible since u belongs to the argmax. So $u_j = p_j^{\min}$.

To sum up, the only coordinates of u that are not extremal are the ones of the "first case" and can be replaced at will by an extremal value without changing the result S_u. Doing so leads to an element of the argmax that has only extremal coordinates. □

Proof of Lemma 5.8

Let us show that a vector u which does not fit the form stated in the lemma cannot be in the argmax. More precisely, assume that there exists j_1 and j_2 such that $f_{j_1} < f_{j_2}$ but $u_{j_1} = p_{j_1}^{\max}$ and $u_{j_2} = p_{j_2}^{\min}$. Then, using the notations of the previous proof, either $f_{j_1} < S_u$ or $f_{j_2} > S_u$, and in each case the previous proof shows that S_u is not maximal, so u does not belong to the argmax. □

Chapter 6

Sequential model aggregation for production forecasting

This chapter is a joint work with Gilles Stoltz, Charles-Pierre Astolfi, Véronique Gervais-Couplet and Sébastien Da Veiga.
It has been submitted for publication.

In this chapter, we apply aggregation methods to an oil production dataset, consisting in monthly simulations and observations of several petrophysical properties over ten years. Classical procedures for oil production forecasting ("history matching") require time-consuming re-computations to take into account intermediate observations along time; our goal is to decrease substantially this computation time by aggregating the simulations results and changing only the weights each month.

We first apply three one-step-ahead individual sequences algorithms: the EWA algorithm, the Ridge regression and the LASSO regression. They give good results, being usually competitive (and even better for LASSO) with the best simulation at hand.

We then apply and adapt the methodology of Chapter 5 to provide forecast intervals for this dataset. At the end, we define a performance measure and a benchmark for the forecast intervals, which we compare with our results.

6.1. Introduction

To optimize the development of a reservoir, engineers need to forecast its production in response to potential development plans. To that purpose, numerical representations – or models – of the reservoir can be considered. They consist of a grid reproducing the structure of the reservoir and populated with facies and petrophysical properties. A fluid-flow simulator is then used to assess the dynamic evolution of the fluids in response to the production scheme. The main difficulty consists in identifying models that are representative of the reservoir. Indeed, many properties of the reservoir cannot be directly related to measurements, and strong uncertainties exist. They can be related to the structure of the reservoir, such as fault location and throw. The reservoir grid can be composed of millions of grid blocks, in which petrophysical properties are mostly unknown. To populate this grid, facies and petrophysical properties are generally considered as random functions characterized by statistical properties inferred from static data such as logs and seismic. Geostatistical approaches are then applied to generate distributions of these properties conditioned to the static data.

Time-dependent measurements, referred to as dynamic data, are also considered to build the reservoir models. They are acquired during the production period, and include measurements at wells such as pressure, oil and water rates, or 4D-seismic related attributes. However, these data are not linearly related to the reservoir properties, so that constraining models to dynamic data is generally a challenging task. Assessing the validity of a model requires to simulate its dynamic behavior. However, fluid-flow simulations can be very long, up to several hours, and the number of potential uncertain parameters very large. Constraining reservoir models to dynamic data, the history-matching process, is thus generally particularly time-consuming. Several methods have been investigated to solve this inverse problem. The variational approach consists in applying minimization algorithms to reduce the objective function that quantifies the error between the production data and the corresponding simulated properties (see Tarantola [1987]). Ensemble methods can also be considered, such as the Ensemble Kalman Filter: cf. Aanonsen et al. [2009]. Interested readers can refer to Oliver and Chen [2011], for instance, for a review of history-matching approaches. One of the main challenges consists in properly parameterizing the problem. Indeed, considering all potential uncertain parameters, such as petrophysical distributions in all grid blocks, would lead to over-parameterization. In addition, the geological consistency of the model needs to be preserved. Several parameterization techniques have thus been proposed in the literature, which aim at mitigating the ill-posedness of the problem while preserving the geological realism. A review of parameterization techniques for petrophysical properties are proposed in Oliver and Chen [2011], Vo and Durlofsky [2016], for instance. However, the history-matching problem remains challenging, especially when discrete properties are considered.

The identification of models constrained to dynamic data is not necessarily the end-goal of reservoir modelling. Indeed, testing potential development scenarios and predicting the corresponding uncertainty on future production generally appears as a crucial step, especially to help decision-making (Scheidt et al. [2015]). Based on this observation, some authors recently focused on the generation of production forecasts constrained to dynamic data without explicitly generating the corresponding updated reservoir models: see Satija and Caers [2015], Satija et al. [2017], Scheidt et al. [2015], Sun and Durlofsky [2017]. These approaches rely on

a Bayesian framework, and use a set of reservoir models to represent prior uncertainty. Fluid-flow simulations performed for this ensemble provide a sampling of the data variables and prediction variables, corresponding to the values simulated for the measured dynamic properties during the history-matching and prediction periods, respectively. The Prediction-Focused Approach (PFA) introduced in Scheidt et al. [2015] consists in applying a dimensionality reduction technique, namely the non-linear principal component analysis (NLPCA), to the two ensembles of variables (data and prediction). The statistical relationship estimated between the two sets of reduced-order variables is then used to estimate the posterior distribution of the prediction variables constrained to the observations using a Metropolis sampling algorithm. This approach is extended in Satija and Caers [2015] to Functional Data Analysis. A Canonical Correlation Analysis is also considered to linearize the relationship between the data and prediction variables in the low-dimensional space. This additional step makes it possible to sample the posterior distribution of prediction variables using simple regression techniques. The resulting approach was demonstrated on a real field case in Satija et al. [2017]. In Sun and Durlofsky [2017], the data and prediction variables are considered jointly in the Bayesian framework. They are first parameterized using Principal Component Analysis (PCA) combined to some mapping operation that aims at reducing the non-Gaussianity of the reduced-order variables. A randomized maximum likelihood algorithm is then used to sample the distribution of the variables given the observed data.

Our approach

The approach we follow also aims at the generation of production forecasts constrained to dynamic data without explicitly generating the corresponding updated reservoir models. Its distinguishing feature with respect to the approaches mentioned above is that it does not rely on a Bayesian framework: it actually relies on no stochastic modeling at all, which actually is in strong contrast with any forecasting method for reservoir production.

This approach uses as building blocks an ensemble of base geological models, from which production forecasts are generated. These models are not updated over time via history-matching, they are simulated once for all. The ensembles of forecasts issued by these models quantify in some sense some uncertainty (the larger the convex hull of forecasts, the more uncertain); see Figure 6.2. Now, a machine-learning algorithm combines these base forecasts at each prediction step, by using convex or linear weights set based on past performance of each base model. This dependency on past performance is where something with a flavor of history-matching is performed. These machine-learning algorithms are called aggregation algorithms. We use them "from the book", with no tweak or adjustment that would be specific to the case of reservoir production, at least as far as one-step-ahead predictions are concerned. This is similar to the studies performed for the forecasting of air quality Mauricette et al. [2009], electricity consumption Devaine et al. [2013], Gaillard and Goude [2015], and exchange rates Amat et al. [2016]. The book Cesa-Bianchi and Lugosi [2006] is an excellent introduction to this sub-field of machine learning called, among other names, prediction with expert advice.

The only methodological modification of this well-established methodology actually consists in an addition: we extend it to not only provide one-step-ahead predictions but also

Table 6.1: Petrophysical properties of the Brugge field

Formation	Depositional environment	Average thickness (m)	Average Porosity	Average Permeability (mD)	Average Net to Gross
Schelde	Fluvial	10	0.207	1105	60
Waal	Lower shoreface	26	0.19	90	88
Maas	Upper shoreface	20	0.241	814	97
Schie	Sandy shelf	5	0.194	36	77

longer-term predictions, as is typically done when forecasting reservoir production. See Section 6.3.2 for details.

Outline of this article

Section 6.2 discusses the (artificial) data set used. Section 6.3 contains a high-level exposition of the machine-learning approach followed, with some additional technical details on the implementations of the algorithms being provided in appendix. Section 6.4 discusses our one-step-ahead predictions while Section 6.5 shows our longer-term predictions. These two sections use two specific machine-learning algorithms, called the ridge regression and the exponentially weighted average forecaster. A final section–Section 6.6—studies a third algorithm, called the LASSO, that not only aggregates the model forecasts but also first selects a subset of the models on which aggregation is then to be performed.

6.2. Brugge case

To assess the potential of the proposed approach for reservoir engineering, we consider the Brugge case, defined by TNO for benchmark purposes: see Peters et al. [2010]. This field, inspired by North Sea Brent-type reservoirs, has an elongated half-dome structure with a modest throw internal fault as shown in Figure 6.1. Its dimensions are about 10km $\times$ 3km. It consists of four main reservoir zones, namely Schelde, Waal, Maas and Schie. The formations with good reservoir properties, Schelde and Maas, alternate with less permeable regions. The average values of the petrophysical properties in each formation are given in Table 6.1. The reservoir is produced by 20 wells located in the top part of the structure. They are indicated in black in Figure 6.1. Ten water injectors are also considered. They are distributed around the producers, near the water-oil contact (blue wells in Figure 6.1).

6.2.1. Reminders on the units

For the mathematical audience intended, we remind the quantities typically measured or monitored when producing oil and gas, as well as their units. We first have the bottomhole pressure (BHP) at the wells, which is measured in pounds per square inch—psi in short. Wells can be of two types, injectors (I) or producers (P). For the latter only, the oil and water production are measured in terms of flow rates, assessed in bbl/day, where bbl is an

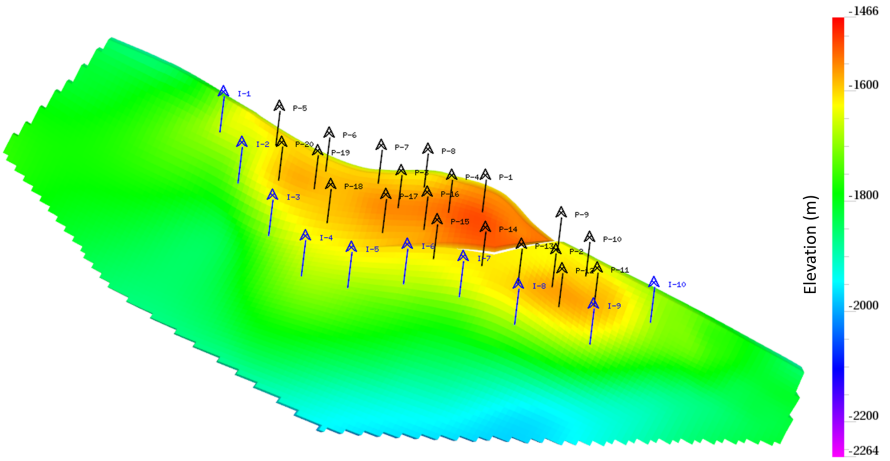


Figure 6.1: Structure of the Brugge field and well location. Producers are indicated in black and injectors in blue.

abbreviation for oilfield barrel, a volume of 42 US gallons, that is, 0.16 m^3. The flow rates of oil, respectively, water, are denoted by q_o (or QO on our pictures), respectively, q_w (or QW).

6.2.2. Description of the construction of the 104 models considered

A fine-scale reference geological model of 20 million grid blocks was initially generated and populated with properties. It was then upscaled to a 450 000 grid block model used to perform the fluid-flow simulation considered in the following as the reference one. The reservoir is initially produced under primary depletion. The producing wells are opened successively during the first 20 months of production. They are imposed a target production rate of 2 000 bbl/day, with a minimum bottomhole pressure of 725 psi. Injection starts in a second step, once all producers are opened. A target water injection rate of 4 000 bbl/day is imposed to the injectors, with a maximal bottomhole pressure of 2 611 psi. A water-cut constraint of 90% is also considered at the producers.

Static data extracted from the reference case were used to generate 104 geological models of 139 $\times$ 48 $\times$ 9 grid blocks ($\sim$ 60 000) provided to the project participants. These models were built considering various approaches for the simulation of facies, fluvial reservoir zones, porosity and permeability. They differ from the distribution of facies, porosity, net-to-gross, water saturation and permeability. More details can be found in Peters et al. [2010], together with examples of permeability realizations.

The dynamic data provided for the benchmark are oil and water rates at the producers plus bottomhole pressure at all wells during a period of 10 years. They were obtained by adding some noise to the results of the reference fluid-flow simulation.

In what follows, the 104 models are used to represent the prior geological uncertainty, see Figure 6.2. (The letter codes explaining which outputs are shown are detailed below.) Two-phase flow simulations are performed for each of them, considering the same production constraints as for the reference case. Additional production constraints could also be consid-

Table 6.2: Summary of the 70 times-series to be predicted.

Code	Well type	Property measured	Units	Number
BHP_Ii	injector	bottomhole pressure	psi	10
BHP_Pj	producer	bottomhole pressure	psi	20
QO_Pj	producer	flow rate of oil	bbl/day	20
QW_Pj	producer	flow rate of water	bbl/day	20

ered, using the production history of the reference case. However, our objective here was to assess the robustness of the proposed approach.

6.2.3. Summary of the 70 properties to be predicted

We respectively index injector wells by $i \in \{1, \ldots, 10\}$ and producer wells by $j \in \{1, \ldots, 20\}$. Table 6.2 summarizes the 70 time-series to be predicted. They will be referred to by codes of the form QO_P19 or BHP_I2 in the following.

Table 6.3 also provides some descriptive statistics pertaining to their orders of magnitude. They should be put in perspective with the root mean-square errors calculated later in Section 6.4.2. In this table, we report both descriptive statistics for the original (nominal) time-series, as well as for the time-series of unit changes[1] (variations between two prediction steps). The latter are the most interesting ones in our view, as far as one-step-ahead forecasting is concerned.

6.3. How to combine the forecasts of the 104 models considered

Several paradigms and theories exist in the field of machine learning to combine (aggregate) the forecasts output by a set of models. Some are designed for the batch case, when all data are available and when only one aggregation is to be performed, while others are sequential in nature: aggregation is to be performed on a regular (e.g., monthly) basis. Some of these aggregation techniques deal with stochastic data: the observations to be forecast may be modeled by some stochastic process, with stationarity often required; on the contrary, other techniques work on deterministic data and come with theoretical guarantees of performance even when the observations cannot be modeled by a stochastic process. Most often, batch methods require stochasticity of the data while sequential methods may get rid of this assumption. Examples of popular aggregation methods include Bayesian model averaging (batch, stochastic) and random forests (batch, stochastic), as well as robust online aggregation (sequential, deterministic), also known as prediction of individual sequences or prediction with expert advice.

[1] We suppressed the extreme changes caused by an opening or a closing of the well when computing the mean, the median, and the standard deviation of the absolute values of these changes.

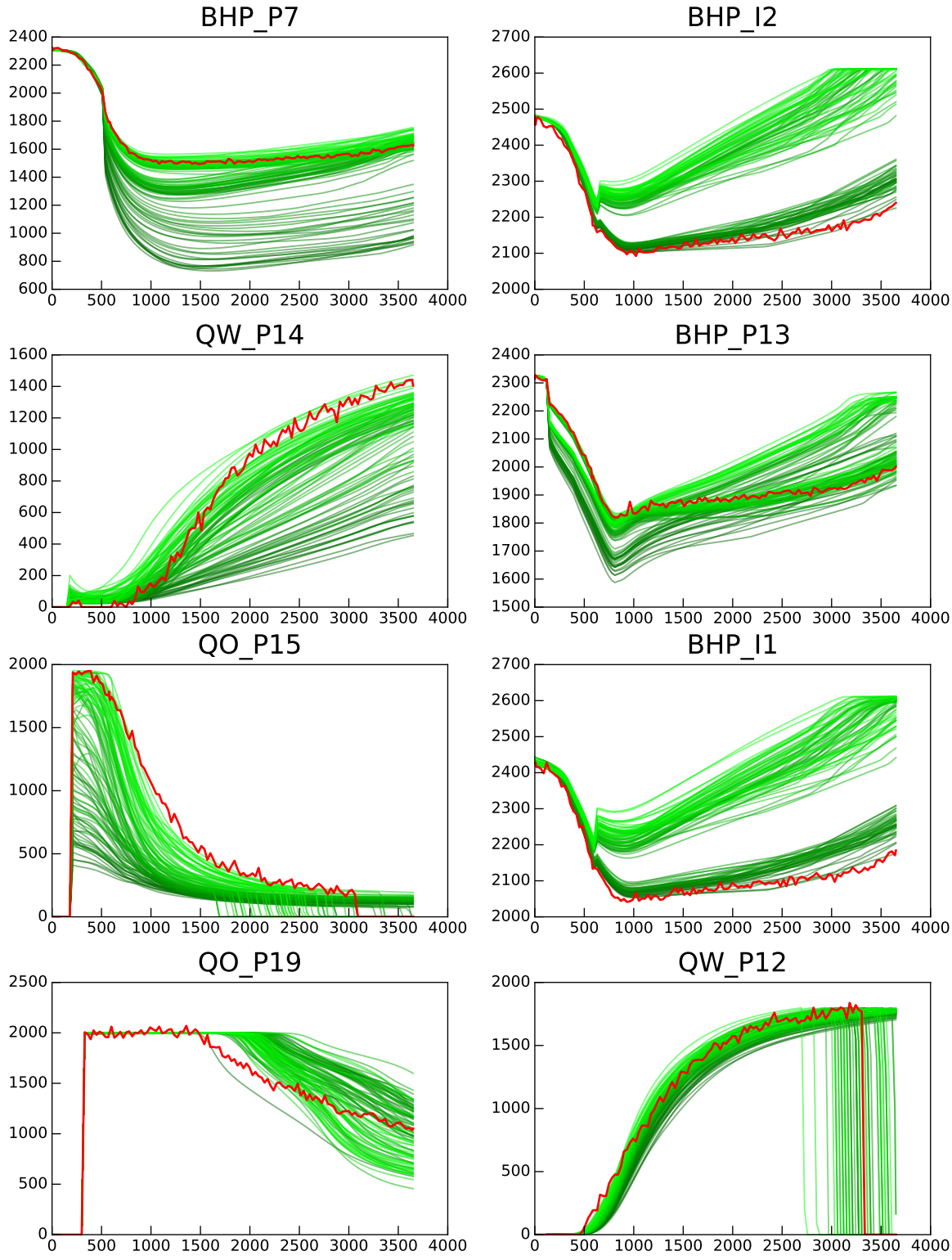


Figure 6.2: The realized production [red] versus the forecasts of the 104 models [green], over time (in days).

Table 6.3: Some descriptive statistics on the properties to be predicted, by type of property. The upper part of the table discusses the nominal time-series, while its lower part studies the time-series of the unit changes.

Property Units	BHP_I psi	BHP_P psi	QO bbl/day	QW bbl/day
Observations				
– Minimum	2 007	708	0	0
– Maximum	2 488	2 380	2 147	1 870
Forecasts output by the models				
– Minimum	1 973	723	0	0
– Maximum	2 610	2 443	2 002	1 800
Unit changes				
– Minimum	−52	−1 308	−306	1 824
– Maximum	39	487	2 147	156
Absolute values of the unit changes				
– Mean	10	12	32	18
– Median	8	9	25	0
– Standard deviation	8	13	28	28

In this article, we will focus on the latter theory, which was developed in the 1990s and is summarized in the monograph Cesa-Bianchi and Lugosi [2006].

6.3.1. High-level methodology: point aggregation for one-step-ahead forecasts

Unless mentioned otherwise, the aggregation of the forecasts will take place well by well, property by property. A first setting is the case of one-step-ahead forecasts.

For a given well and a given property, we denote by $y_1, y_2, \ldots, y_T$ the sequence of the observed values of the property over time and by $m_{j,t}$, where $j \in \{1, \ldots, 104\}$ and $t \in \{1, \ldots, T\}$, the sequence of the sets of forecasts output by the 104 models considered. At each step t, we linearly combine the forecasts $m_{j,t}$ to form our aggregated forecast $\widehat{y}_t$:

$$\widehat{y}_t = \sum_{j=1}^{104} w_{j,t}\, m_{j,t}\,, \tag{6.3.1}$$

where the weights $w_{j,t}$ are determined based on the past observations y_s and past forecasts $m_{j,s}$, where $s \leqslant t-1$. The precise formulae to set these weights (some specific algorithms designed by the literature) are detailed in Section 6.3.3 below. The basic idea is to put higher weights on models that performed better in the past.

The main interest of this methodology is given by its performance guarantee: the weights can be set to mimic the performance of some good constant combination of the forecasts.

More precisely, given a set $\mathcal{W}^\star$ of reference weights (e.g., the set of all convex weights, or of all linear weights in some compact ball), good algorithms ensure that, no matter what the observations y_t and the forecasts $m_{j,t}$ of the models were,

$$\frac{1}{T}\sum_{t=1}^{T}\big(\widehat{y}_t - y_t\big)^2 \leqslant \varepsilon_T + \inf_{(v_1,\dots,v_{104})\in\mathcal{W}^\star} \frac{1}{T}\sum_{t=1}^{T}\left(\widehat{y}_t - \sum_{j=1}^{104} v_j\, m_{j,t}\right)^2, \tag{6.3.2}$$

where ε_T is a small term, typically of order $1/\sqrt{T}$. More details are given in Section 6.3.3, for each specific algorithm.

The reference set $\mathcal{W}^\star$ will always include the weights $(v_1,\dots,v_{104})$ of the form $(0,\dots,0,1,0,\dots,0)$ that only put non-zero mass equal to 1 on one model. Thus, the infimum over elements in $\mathcal{W}^\star$ will always be smaller than the cumulative square loss of the best of the 104 models. For some algorithms, this reference set $\mathcal{W}^\star$ will be much larger and will contain all weights of some Euclidean ball of $\mathbb{R}^{104}$ with radius larger than 1, thus in particular, all convex weights. Note also that no stochastic modeling of the observations or of the forecasts of the models is required; all possible (bounded) sequences can be considered.

The algorithms we will consider (and the way we will refer to them in the sequel) are: the exponentially weighted average (EWA) forecaster; the ridge regression (Ridge); the LASSO regression (LASSO). Their statement and theoretical guarantees—in terms of the quantities $\mathcal{W}^\star$ and ε_T in (6.3.2)—are detailed in Section 6.3.3.

Before we describe them in details, we provide a high-level view on the second aspect of our methodology, pertaining to longer-term predictions.

6.3.2. High-level methodology: longer-term predictions

The point aggregation indicated above is for one-step-ahead predictions: it may be performed as long as properties are measured on a regular basis (in technical words: as long as y_t is measured after outputting $\widehat{y}_t$ and before $\widehat{y}_{t+1}$ is to be output). Denote by T the last step of measurement of the property considered. Note that the geological models still provide forecasts for rounds $t \geqslant T+1$.

Now, we may also be interested in providing interval forecasts for longer-term forecasts, i.e., for rounds $t = T+k$, where $k \geqslant 1$ and k can be possibly large. We determine our interval for round $T+k$ by first determining a set $\widehat{\mathcal{W}}_{T+k}$ where the desirable weights $(w_{1,T+k},\dots,w_{104,T+K})$ lie in; this set of weights depends on the algorithm considered and on its performance and behavior on the T rounds of sequential prediction. The interval forecast for round $t = T+k$ is then

$$\widehat{S}_{T+k} = \mathrm{conv}\left\{\sum_{j=1}^{104} w_{j,T+k}\, m_{j,T+k} : \quad (w_{1,T+k},\dots,w_{104,T+k}) \in \widehat{\mathcal{W}}_{T+k}\right\}, \tag{6.3.3}$$

where conv denotes a convex hull, possibly with some enlargement to take into account the noise level.

The question is how to determine these sets $\widehat{\mathcal{W}}_{T+k}$ of desirable weights. Figure 6.3 illustrates the methodology:

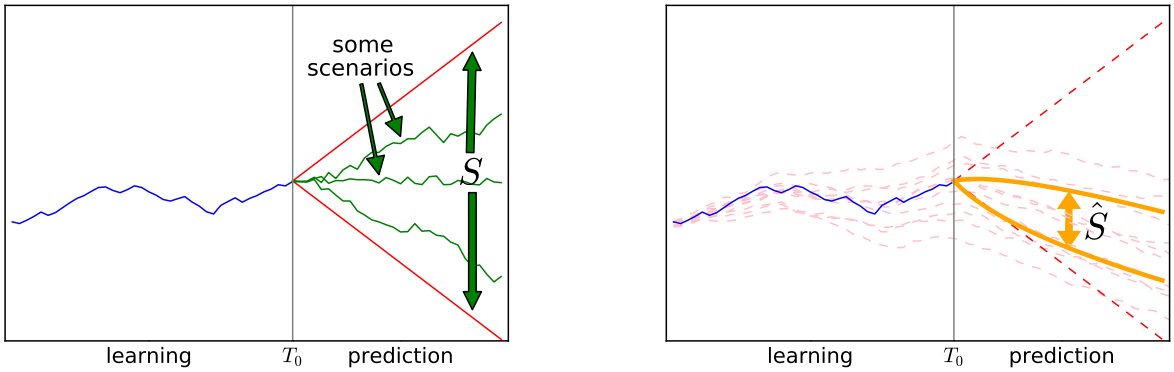


Figure 6.3: Schematic diagram for interval forecasting.

1. On the first part of the data set (called the learning or training part), we use the algorithms as explained above for point aggregation.

2. On the second part of the data set (called the prediction part),

   - we consider the set $S = S_{T+1} \times \ldots \times S_{T+k} \times \ldots$ of all plausible continuations $z_{T+1}, \ldots, z_{T+k}, \ldots$ of the observations $y_1, \ldots, y_T$; this set S will be referred to as the set of scenarios;
   - for a given scenario $y_1, \ldots, y_T, z_{T+1}, z_{T+2}, \ldots$, we compute the weights $(w_{1,T+k}, \ldots, w_{104,T+k})$ to use at round $t+k$ by running the considered algorithm on the putative past observations $y_1, \ldots, y_T, z_{T+1}, \ldots, z_{T+k-1}$ and past model forecasts;
   - we form the aggregated forecast $\widehat{z}_{T+k} = \sum_j w_{j,T+k}\, m_{j,T+k}$.

3. The interval forecasts $\widehat{S}_{T+k}$ are the convex hulls of all possible aggregated forecasts $\widehat{z}_{T+k}$ obtained by running all scenarios in S (with possibly some enlargement to take into account the noise level).

The main constructions remaining to be explained is (i) how the set S of plausible continuations is determined; (ii) how we may efficiently compute the interval forecasts $\widehat{S}_{T+k}$, as there are infinitely many scenarios; (iii) what we mean by an enlargement to account for the noise level. We provide all needed explanations in Section 6.7 (which actually mostly refers to Chapter 5 of this thesis).

The high-level idea for (i) is however that we look on available data how large the typical variations were, which yields an interval $[m, M]$ of typical 1-step average variations. The set of scenarios is then the cone formed by the product of the intervals $[y_T + km,\ y_T + kM]$, where $k = 0, 1, 2, \ldots$. See Figure 6.4 for an illustration. As for (ii), we should note that only the upper and lower bounds of $\widehat{S}_{T+k}$ need to be computed (or bounded), which is not too difficult a task.

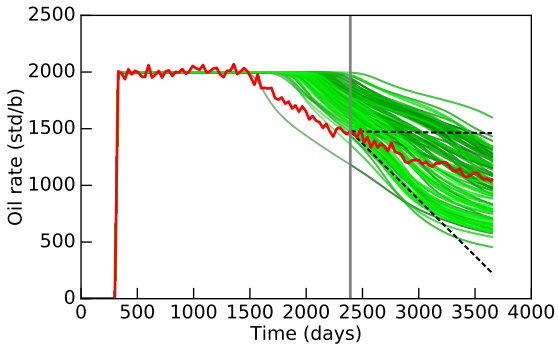


Figure 6.4: An example of the set of scenarios S calculated for QO_P19 on the last third of the sample.

6.3.3. Statement of the aggregation algorithms considered

We provided hereabove the general methodological framework for sequentially aggregating forecasts in a robust manner, not relying on any stochastic modeling of the observations or of the forecasts of the models. We now provide the statements and the theoretical guarantees of the considered algorithms; the theoretical guarantees refer to (6.3.2) and consist in providing the values of $\mathcal{W}^\star$ and ε_T for the considered algorithm.

We only deal with point aggregation in this subsection. The extension to interval aggregation is briefly discussed in Section 6.7.

For the sake of generality, we write K to denote the number of underlying models when stating our algorithms ($K = 104$ with the Brugge data set).

The ridge regression (Ridge)

The ridge regression (which we will refer to as Ridge when reporting the experimental results) was introduced by Hoerl and Kennard [1970] in a stochastic and non-sequential setting. What follows relies on recent new analyses of the ridge regression in the machine learning community; see the original papers by Azoury and Warmuth [2001], Vovk [2001] and the survey in the monograph by Cesa-Bianchi and Lugosi [2006], as well as the discussion and the optimization of the bounds found in these references proposed by Gerchinovitz [2011].

Ridge relies on a parameter $\lambda > 0$, called a regularization factor. At round $t = 1$, it picks arbitrary weights, e.g., uniform $(1/K, \dots, 1/K)$ weights. At rounds $t \geqslant 2$, it picks

$$(w_{1,t}, \dots, w_{K,t}) \in \operatorname*{argmin}_{(v_1,\dots,v_K)\in\mathbb{R}^K} \left\{ \lambda \sum_{j=1}^{K} v_j^2 + \sum_{s=1}^{t-1} \left(y_s - \sum_{j=1}^{K} v_j \, m_{j,s} \right)^2 \right\}; \tag{6.3.4}$$

i.e., it picks the best constant weights to reconstruct past observations based on the model forecasts subject to an ℓ^2–regularization constraint $\sum v_j^2$, which is useful to avoid overfitting to the past.

The performance bound relies on two bounds V and B and is over

$$\mathcal{W}^\star = \left\{ (v_1, \dots, v_K) : \quad \sum_{j=1}^K v_j^2 \leqslant V^2 \right\},$$

the Euclidean ball of $\mathbb{R}^K$ with center $(0, \dots, 0)$ and radius $V \geqslant 1$. This ball contains in particular all convex combinations in $\mathbb{R}^K$. The bound (6.3.2) with the above $\mathcal{W}^\star$ reads: for all bounded sequences of observations $y_t \in [-B, B]$ and model forecasts $m_{j,t} \in [-B, B]$,

$$\varepsilon_T \leqslant \frac{1}{T} \left(\lambda V^2 + 4KB^2 \left(1 + \frac{KB^2 T}{\lambda} \right) \ln \left(1 + \frac{B^2 T}{\lambda} \right) + 5B^2 \right).$$

In particular, for a well-chosen λ of order $\sqrt{T}$, we have $\varepsilon_T = O\big((\ln T)/\sqrt{T}\big)$.

The latter choice on λ depends however on the quantities T and B, which are not always known in advance. This is why in practice we set the λ to be used at round t based on past data. More explanations and details are provided below.

The LASSO regression (LASSO)

The LASSO regression was introduced by Tibshirani [1996], see also the efficient implementation proposed in Efron et al. [2004]. Its definition is similar to the definition (6.3.4) of Ridge, except that the ℓ^2–regularization is replaced by an ℓ^1–regularization: at rounds $t \geqslant 2$,

$$(w_{1,t}, \dots, w_{K,t}) \in \operatorname*{argmin}_{(v_1, \dots, v_K) \in \mathbb{R}^K} \left\{ \lambda \sum_{j=1}^K |v_j| + \sum_{s=1}^{t-1} \left(y_s - \sum_{j=1}^K v_j \, m_{j,s} \right)^2 \right\}.$$

As can be seen from this definition, LASSO also relies on a regularization parameter $\lambda > 0$.

One of the key features of LASSO is that the weights $(w_{1,t}, \dots, w_{K,t})$ it picks are often sparse: many of its components are null. Unfortunately, we are not aware of any performance guarantee of the form (6.3.2): all analyses of LASSO we know rely on (heavy) stochastic assumptions and are tailored to non-sequential data. We nonetheless implemented it and tabulated its performance, as well as its selection power (since the weights are sparse, many models are discarded). See Section 6.6 for details.

The exponentially weighted average (EWA) forecaster

The previous two forecasters were using linear weights: weights that lie in $\mathbb{R}^K$ but are not constrained to be nonnegative or to sum up to 1. In contrast, the exponentially weighted average (EWA) forecaster picks convex weights: weights that are nonnegative and sum up to 1. The aggregated forecasts $\widehat{y}_t$ lie therefore in the convex hull of the forecasts $m_{j,t}$ of the models, which may be considered a safer way to predict.

EWA (sometimes called Hedge) was introduced by Littlestone and Warmuth [1994], Vovk [1990] and further understood and studied by, among others, Auer et al. [2002], Cesa-Bianchi et al. [1997], Cesa-Bianchi [1999]; see also the monograph by Cesa-Bianchi and Lugosi [2006].

This algorithm picks uniform $(1/K, \dots, 1/K)$ weights at round $t = 1$, while at subsequent rounds $t \geqslant 2$, it picks weights $(w_{1,t}, \dots, w_{K,t})$ such that

$$w_{j,t} = \frac{\exp\left(-\eta \sum_{s=1}^{t-1} (y_s - m_{j,s})^2\right)}{\sum_{k=1}^{K} \exp\left(-\eta \sum_{s=1}^{t-1} (y_s - m_{k,s})^2\right)}.$$

The weight put on model j at round t depends on the cumulative accuracy error suffered by j on rounds 1 to $t-1$; however, the weight is not directly proportional to this cumulative error: a rescaling via the exponential function is operated, with a parameter $\eta > 0$. We will call this parameter the learning rate of EWA: when η is smaller, the weights get closer to the uniform weights; when η is larger, the weights of the suboptimal models get closer to 0 while the (sum of the) weight(s) of the best-performing model(s) on the past get closer to 1.

To provide the performance bound we first denote by δ_j the convex weight vector $(0, \dots, 0, 1, 0, \dots, 0)$, where the unique non-zero coordinate is the j–th one. The set $\mathcal{W}^\star$ of reference weights is given by

$$\mathcal{W}^\star = \left\{\delta_j : \quad j \in \{1, \dots, K\}\right\}.$$

The performance bound (6.3.2) with the above $\mathcal{W}^\star$ relies on a boundedness parameter B and reads: for all bounded sequences of observations $y_t \in [0, B]$ and model forecasts $m_{j,t} \in [0, B]$,

$$\varepsilon_T \leqslant \begin{cases} \dfrac{\ln K}{\eta T} & \text{if } \eta \leqslant 1/(2B^2), \\ \dfrac{\ln K}{\eta T} + \dfrac{\eta B^2}{8} & \text{if } \eta > 1/(2B^2). \end{cases}$$

In particular, $\varepsilon_T = O(1/T)$ if η is well-calibrated, which requires the knowledge of a plausible bound B. Here again, we may prefer to set the η to be used at round t based on past data; see the next section.

How to implement these algorithms (i.e., pick their parameter λ or η)

First, note that the algorithms described above rely each on a single parameter $\lambda > 0$ or $\eta > 0$, which is in strong contrast with the geophysical models constructed. (These parameters λ and η are actually rather called hyperparameters to distinguish them from the model parameters.)

In addition, the literature provides theoretical or practical guidelines on how to choose these parameters. The key idea was introduced by Auer et al. [2002]. It consist in letting the parameters η or λ vary over time: we denote by η_t and λ_t the parameters used to pick the weights $(w_{1,t}, \dots, w_{K,t})$ at round t. Theoretical studies offer some formulas for η_t and λ_t (see, e.g., Auer et al. [2002], Cesa-Bianchi et al. [2007]) but the associated practical performance are usually poor, or at least, improvable, as noted first by Devaine et al. [2013] and later by Amat et al. [2016]. This is why Devaine et al. [2013] suggested and implemented the following

tuning of η_t and λ_t on past data, which somehow adapts to the data without overfitting; it corresponds to a grid search of the best parameters on available past data.

More precisely, we respectively denote by $\mathcal{R}_\lambda$ and $\mathcal{E}_\eta$ the algorithms Ridge run with constant regularization factor $\lambda > 0$ and EWA run with constant learning rate $\eta > 0$. We further denote by L_{t-1} the cumulative loss they suffered on prediction steps $1, 2, \ldots, t-1$:

$$L_{t-1}(\,\cdot\,) = \sum_{s=1}^{t-1} (\widehat{y}_s - y_s)^2 \,,$$

where the $\widehat{y}_s$ denote the predictions output by the algorithm considered, $\mathcal{R}_\lambda$ or $\mathcal{E}_\eta$. Now, given a finite grid $\mathcal{G} \subset (0, +\infty)$ of possible values for the parameters λ or η, we pick, at round $t \geqslant 2$,

$$\lambda_t \in \operatorname*{argmin}_{\lambda \in \mathcal{G}} \big\{ L_{t-1}(\mathcal{R}_\lambda) \big\} \qquad \text{and} \qquad \eta_t \in \operatorname*{argmin}_{\eta \in \mathcal{G}} \big\{ L_{t-1}(\mathcal{E}_\eta) \big\} \,,$$

and then form our aggregated prediction $\widehat{y}_t$ for step t by using either the aggregated forecast output by $\mathcal{R}_{\lambda_t}$ or $\mathcal{E}_{\lambda_t}$.

We resorted to wide grids in our implementations, as the various properties to be forecast have extremely different orders of magnitude:

- for EWA, 300 equally spaced points in logarithmic scale between 10^{-20} and 10^{10};
- for LASSO, 100 such points between 10^{-20} and 10^{10};
- for Ridge, 100 such between 10^{-30} and 10^{30}.

However, how fine the grids are has not a significant impact on the performance; what matter most is that the correct orders of magnitude for the hyperparameters be covered by the considered grids.

6.4. Results of point aggregation for one-step-ahead forecasts

We discuss here the application of the Ridge and EWA algorithms on the Brugge data set. The latter covers 10 years of time but we only consider 127 evenly spaced prediction steps, which are thus roughly separated by a month. The one-step-ahead predictions discussed in this section thus (roughly) correspond to one-month-ahead predictions.

6.4.1. Qualitative study on a few representative properties

Figure 6.5 reports the forecasts of Ridge and EWA for 8 selected properties (which are representative of the 70 properties to be predicted).

The main comment would be that the aggregated forecasts look close enough to the true observations, even though most of the model forecasts they build on may err. The aggregated forecasts evolve typically in a smoother way than the observations. The right-most part of the BHP_I1 picture reveals that Ridge is typically less constrained than EWA by the ensemble of forecasts: while EWA cannot provide aggregated forecasts that are out of the convex hull

of the ensemble forecasts, Ridge resorts to linear combinations of the latter and may thus output predictions out of this range.

Also, the middle part of the QO_P19 picture reveals that Ridge typically adjusts better and faster to regime changes, while EWA takes some more time to depart from a simulation it had sticked to. The disadvantage of this for Ridge is that it may at times react too fast: see the bumps in the initial time steps on pictures BHP_P12 and QW_P18.

6.4.2. Quantitative study on all properties

A more objective assessment of the performance of Ridge and EWA can be obtained through an accuracy criterion. As is classical in the literature, we resort to the root mean-square error (RMSE), which we define by taking into account a training period: by taking out of the evaluation the first 31 predictions (that is, roughly 1/4 of the observations). Hence, the algorithms are only evaluated on time steps 32 to 127, as follows:

$$\frac{1}{127-32+1}\sum_{t=32}^{127}\left(\widehat{y}_t - y_t\right)^2.$$

Similar formulae determine on this time interval 32–127 the performance of the best model and of the best convex combination of the models:

$$\min_{j=1,\dots,104} \frac{1}{127-32+1}\sum_{t=32}^{127}(m_{j,t}-y_t)^2$$

$$\text{and} \qquad \min_{(v_1,\dots,v_{104})\in\mathcal{C}} \frac{1}{127-32+1}\sum_{t=32}^{127}\left(\sum_{j=1}^{104} v_j\, m_{j,t} - y_t\right)^2$$

where $\mathcal{C}$ denotes the set of all convex weights, i.e., all vectors of $\mathbb{R}^{104}$ with non-negative coefficients summing up to 1. The best model and the best convex combination of the models vary by properties; this is why we will sometimes write the "best local model" or the "best local convex combination".

Note that the orders of magnitude of the properties are extremely different, depending on what is measured (they tend to be similar within a given category). We did not correct for that and did not try to normalize the RMSEs. (Considering other criteria like the mean absolute percentage of error—MAPE—would help to get such a normalization.)

The various RMSEs introduced above are represented in Figure 6.6. A synthetical summary of performance would be that Ridge typically gets an overall accuracy close to that of the best local convex combination while EWA rather performs like the best local model. This is perfectly in line with the theoretical guarantees described in Section 6.3.3. But Ridge has a drawback: the instabilities (the reactions that might come too fast) already underlined in our qualitative assessment result in a few absolutely disastrous performance, in particular for BHP_P5, BPH_P10, QW_P16, QW_P12. The EWA algorithm seems a safer option, though not being as effective as Ridge. The deep reason why EWA is safer comes from its definition: it only resorts to convex weights of the model forecasts, and never predicts a value larger (smaller) than the largest (smallest) forecast of the models.

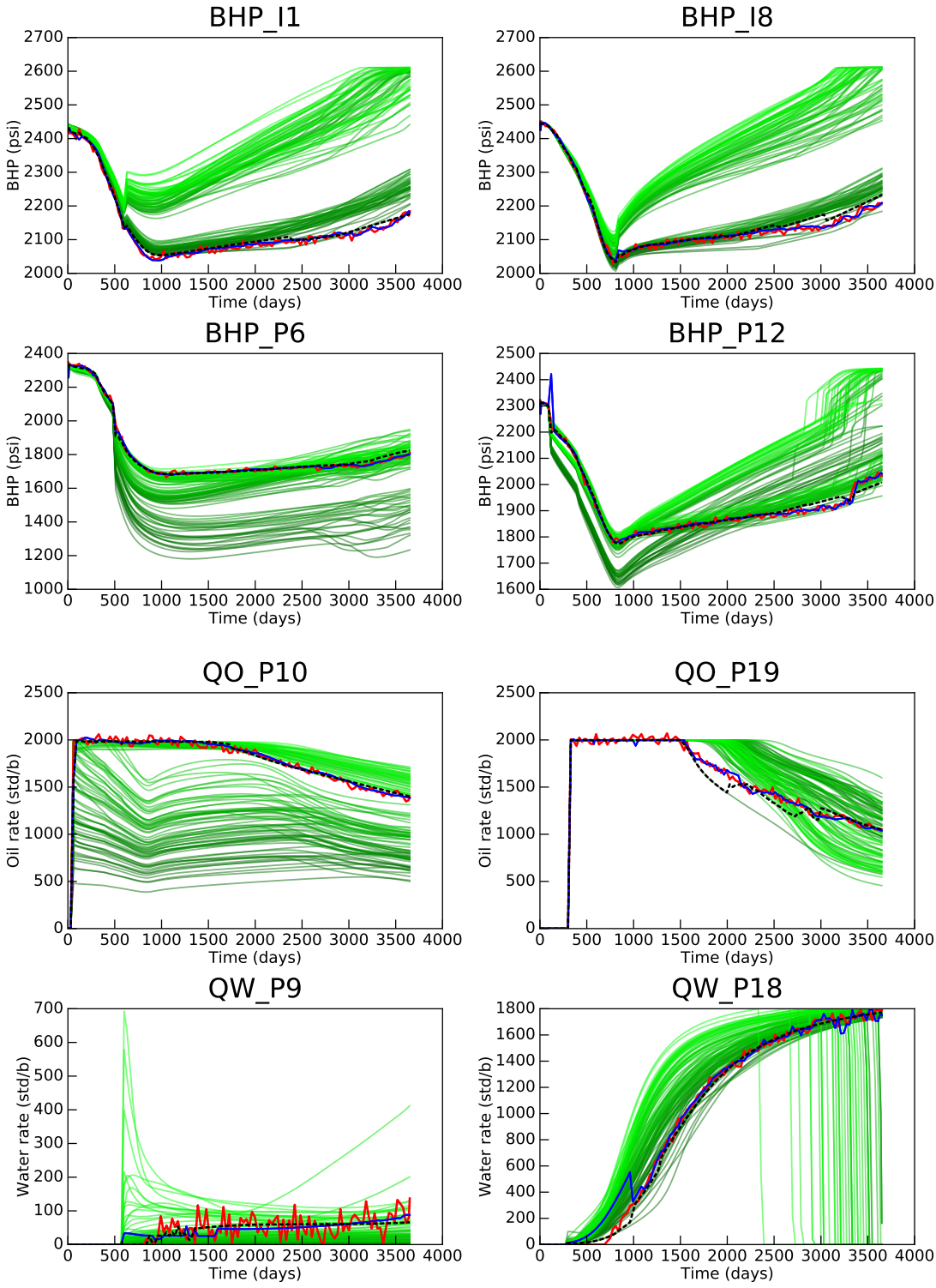
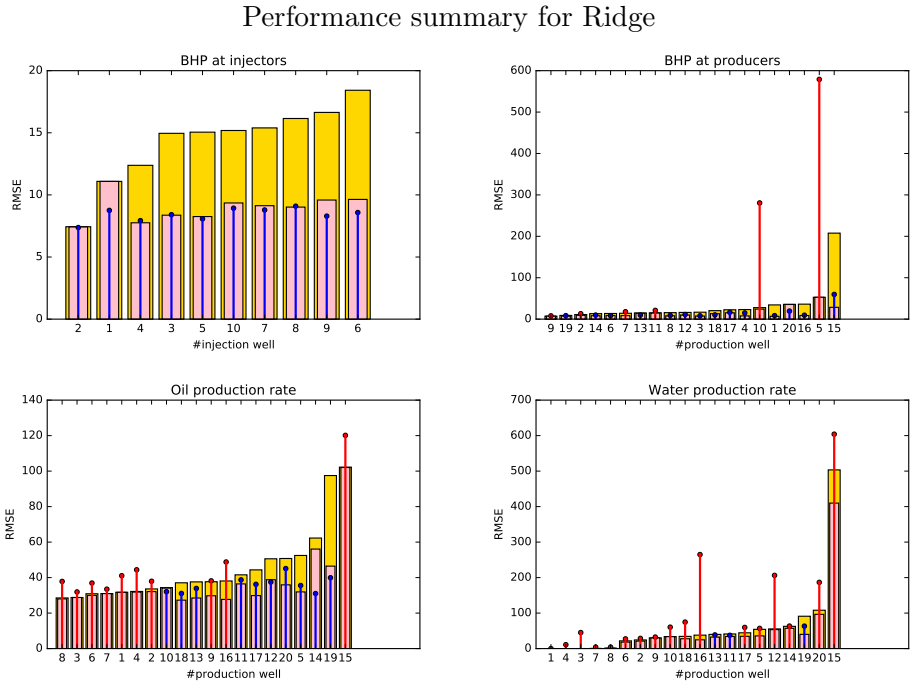


Figure 6.5: Simulations of the models (green lines —), observations (red solid line —), one-step-ahead forecasts by Ridge (blue solid line —) and EWA (black dotted line - - -).



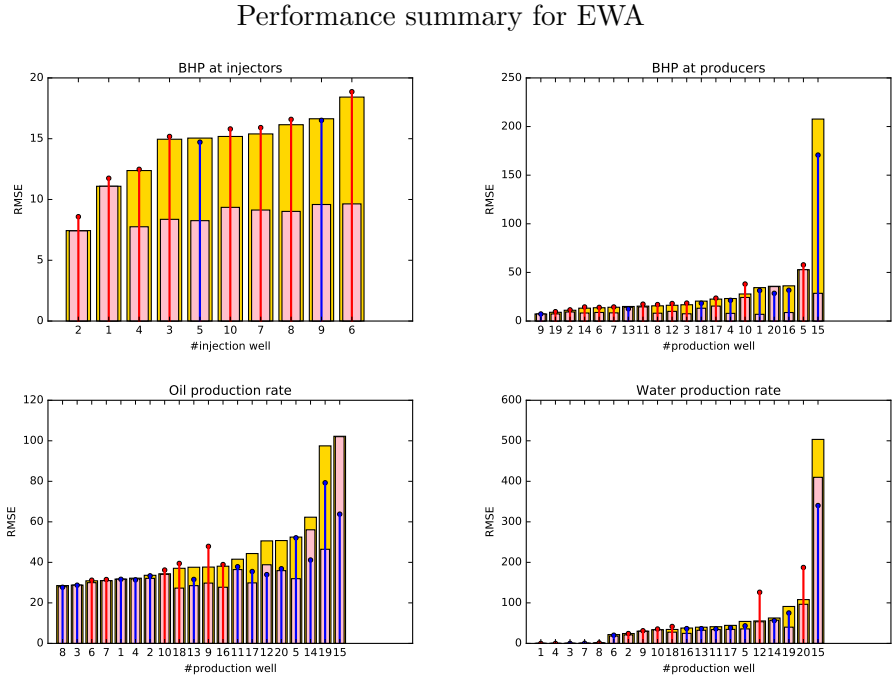


Figure 6.6: RMSEs of the best model (yellow bars) and of the best convex combination of models (pink bars) for each property, as well as the RMSEs of the considered algorithms: Ridge (top graphs) and EWA (bottom graphs). The RMSEs of the latter are depicted in blue whenever they are smaller than that of the best model for the considered property, in red otherwise.

6.5. Results for interval aggregation

In this section we again discuss only the application of the Ridge and EWA algorithms on the Brugge data set. We use the first two thirds of the observations (times steps 1 to 84) as a training sample. Based on these observations, the methods considered output longer-term forecasts in the form of prediction intervals as explained in Section 6.3.2. They do so for the prediction steps 85 to 127. (We note that Ridge selects first a sub-sample of the simulations to build these interval forecasts, see Section 6.7.2 for details.)

We now provide a qualitative assessment of the interval forecasts obtained.

6.5.1. Some good results

Figures 6.7 and 6.8 report interval forecasts that look good: they are significantly narrower than the sets of scenarios while containing most of the observations.

They were obtained, though, by using some hand-picked parameters λ or η: we manually performed some trade-off between the widths of the interval forecasts (which is expected to be much smaller than the set S of all scenarios) and the accuracy of the predictions (a large proportion of future observations should lie in the interval forecasts).

We were unable so far to get any totally satisfactory automatic tuning of these parameters (however, two attempts are presented in Section 6.8.3, with interesting results: Section 6.8.4). Hence, the good results achieved on Figures 6.7 and 6.8 merely hint at the potential benefits of our methods once they will come with proper parameter-tuning rules.

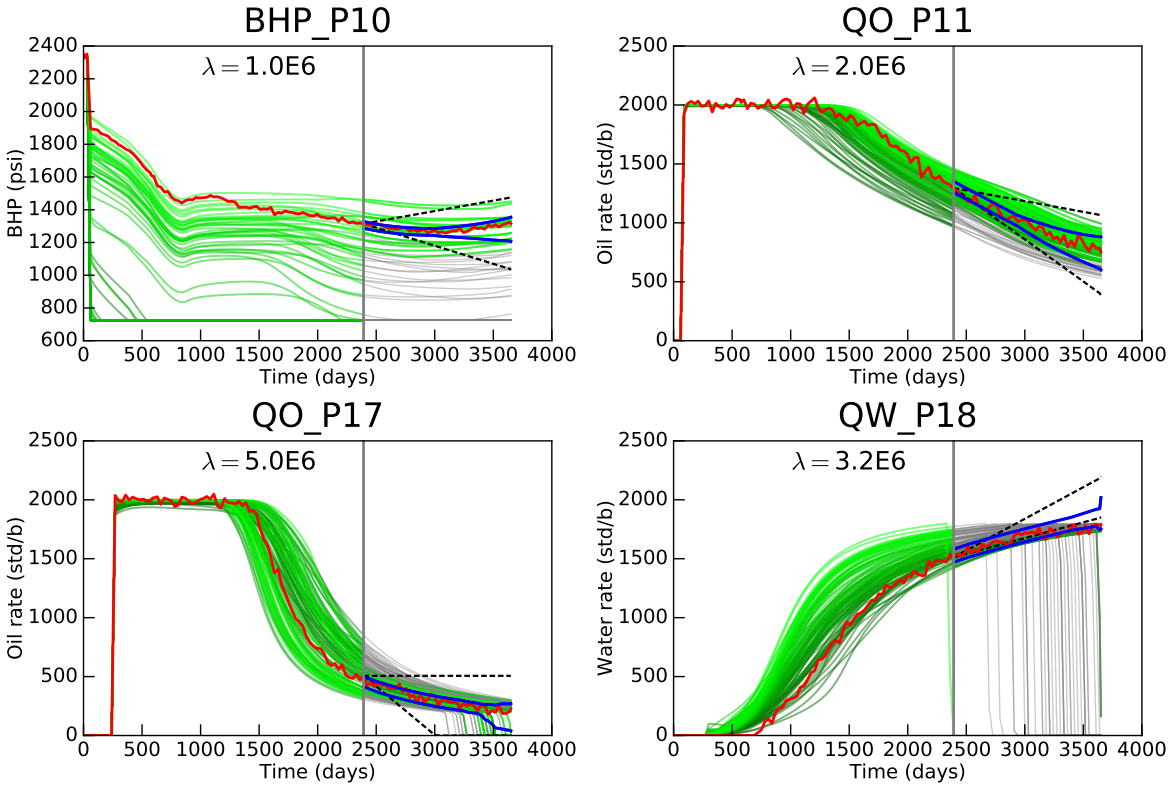


Figure 6.7: Simulations of the models (green lines — or grey lines —, depending on whether the simulations were selected for the interval forecasts), observations (red solid line —), set S of scenarios (upper and lower bounds given by black dotted lines - - -), and interval forecasts output by Ridge (upper and lower bounds given by blue solid lines —). Values of λ used are written on the graphs.

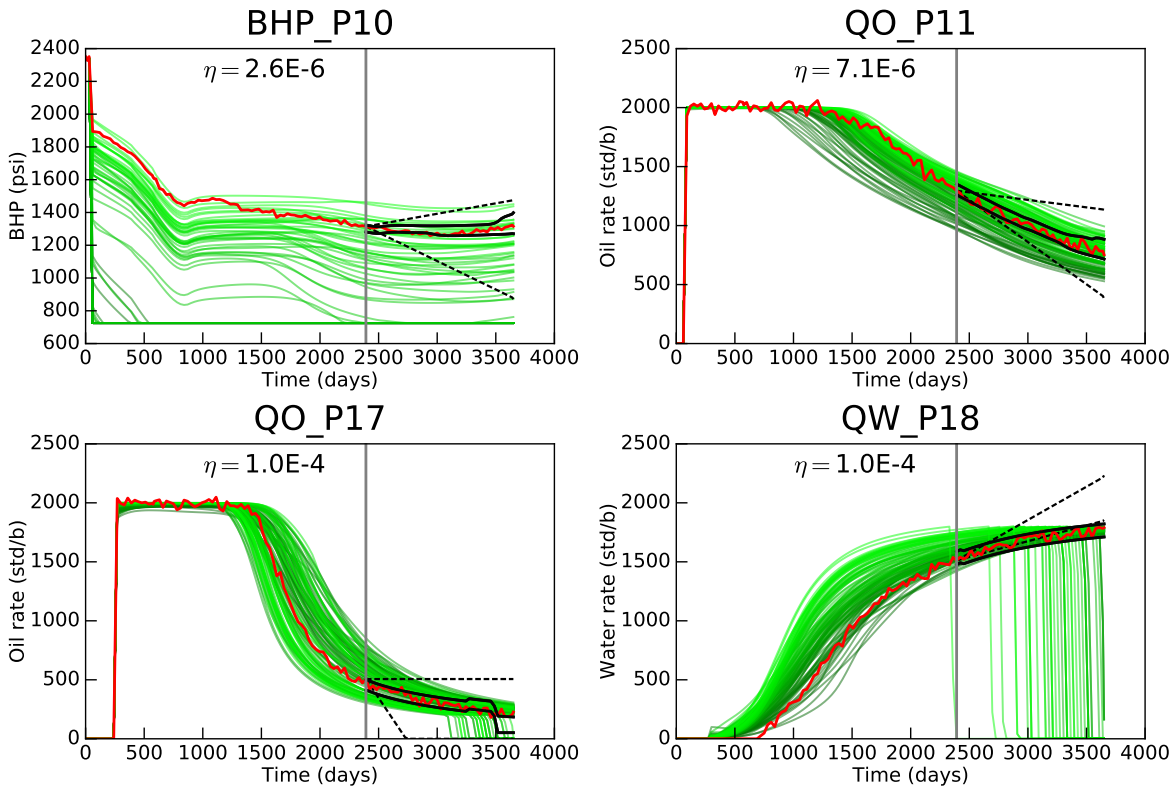


Figure 6.8: Simulations of the models (green lines —), observations (red solid line —), set S of scenarios (upper and lower bounds given by black dotted lines - - -), and interval forecasts output by EWA (upper and lower bounds given by **solid lines** —). Values of η used are written on the graphs.

6.5.2. Some disappointing results

Figures 6.9 and 6.10 show, on the other hand, that for some properties, neither Ridge nor EWA may provide useful interval forecasts: the latter either completely fail to accurately predict the observations or they are so large that they cover (almost) the set of all scenarios — hence, they do not provide any useful information.

We illustrate this by letting λ increase (Figure 6.9) and η decrease (Figure 6.10): the interval forecasts become larger as the parameters vary in this way. They first provide inaccurate interval forecasts and finally resort to intervals (almost) covering all scenarios.

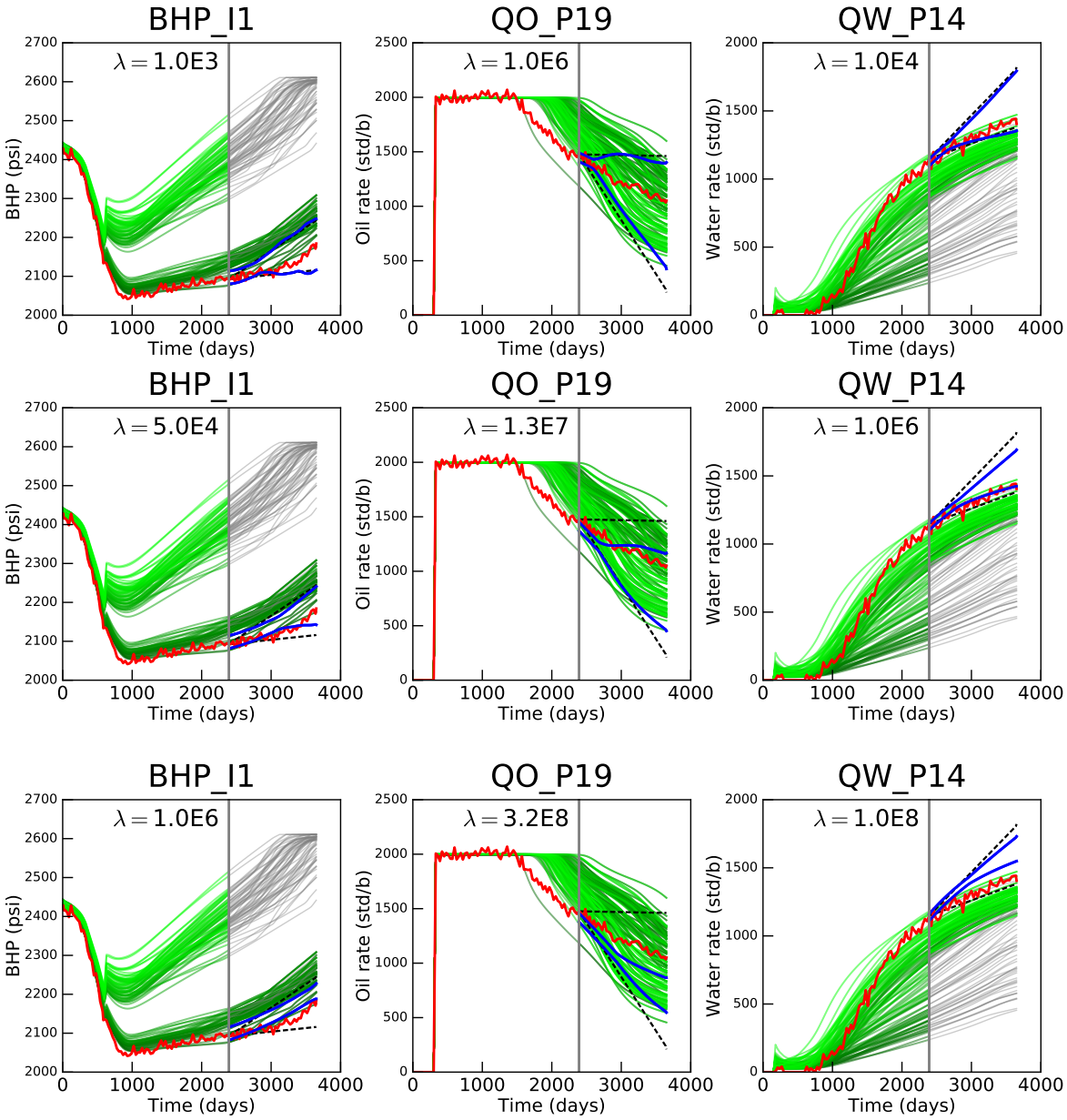


Figure 6.9: Simulations of the models (green lines — or grey lines —, depending on whether the simulations were selected for the interval forecasts), observations (red solid line —), set S of scenarios (upper and lower bounds given by black dotted lines - - -), and interval forecasts output by Ridge (upper and lower bounds given by blue solid lines —). Values of λ used are written on the graphs.

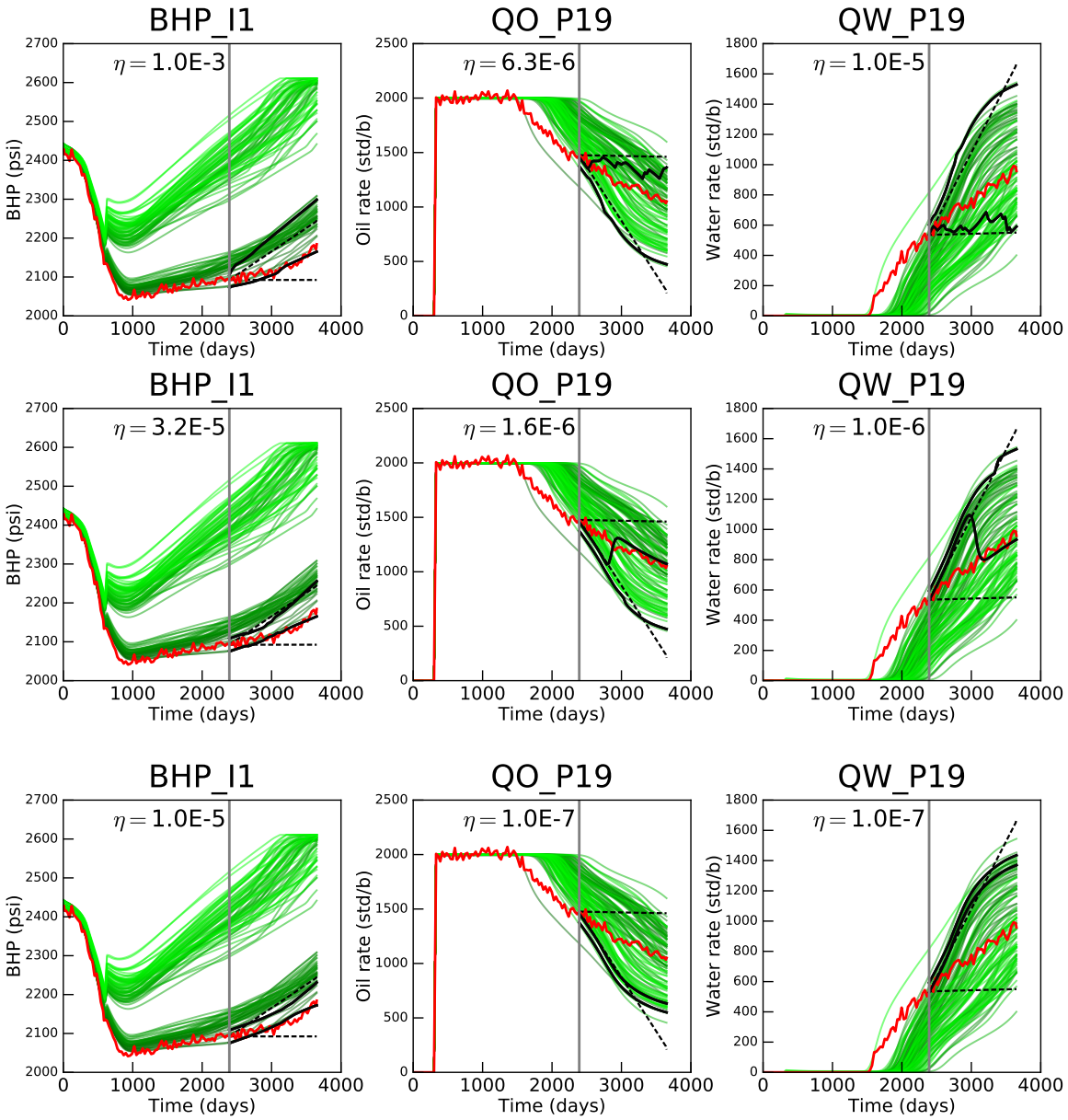


Figure 6.10: Simulations of the models (green lines —), observations (red solid line —), set S of scenarios (upper and lower bounds given by black dotted lines - - -), and interval forecasts output by EWA (upper and lower bounds given by **solid lines** —). Values of η used are written on the graphs.

6.6. LASSO

We conclude our experiments with a discussion of the performance of LASSO. Two points are to be discussed: first, the accuracy of the one-step-ahead forecasts, as we did for EWA and Ridge, and second, the selection power of LASSO.

As far as the forecasting is concerned, we recall that we were only able to produce one-step-ahead forecasts with LASSO, not longer-term interval forecasts. Figures 6.11–6.12 reveal, compared to the similar graphs for Ridge (in particular, Figure 6.6, top), that the accuracy achieved by LASSO is slightly better than that of Ridge, with only one exception, well number 9 in terms of oil production rate. Otherwise, LASSO basically gets the best out of the accuracy of EWA (which predicted well all properties for producers, namely, bottomhole pressure, oil production rate and water production rate) and that of Ridge (which predicted well the bottomhole pressure for injectors).

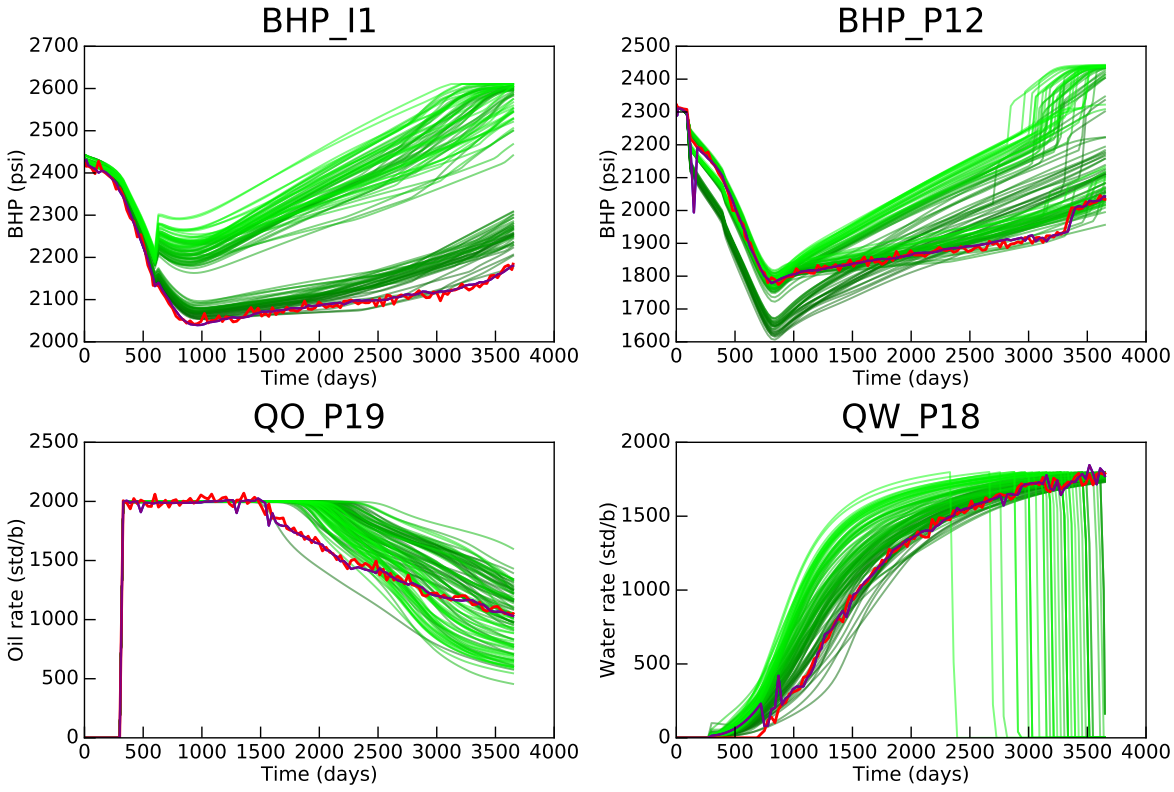


Figure 6.11: Simulations of the models (green lines —), observations (red solid line —) and one-step-ahead forecasts by LASSO (purple solid line —).

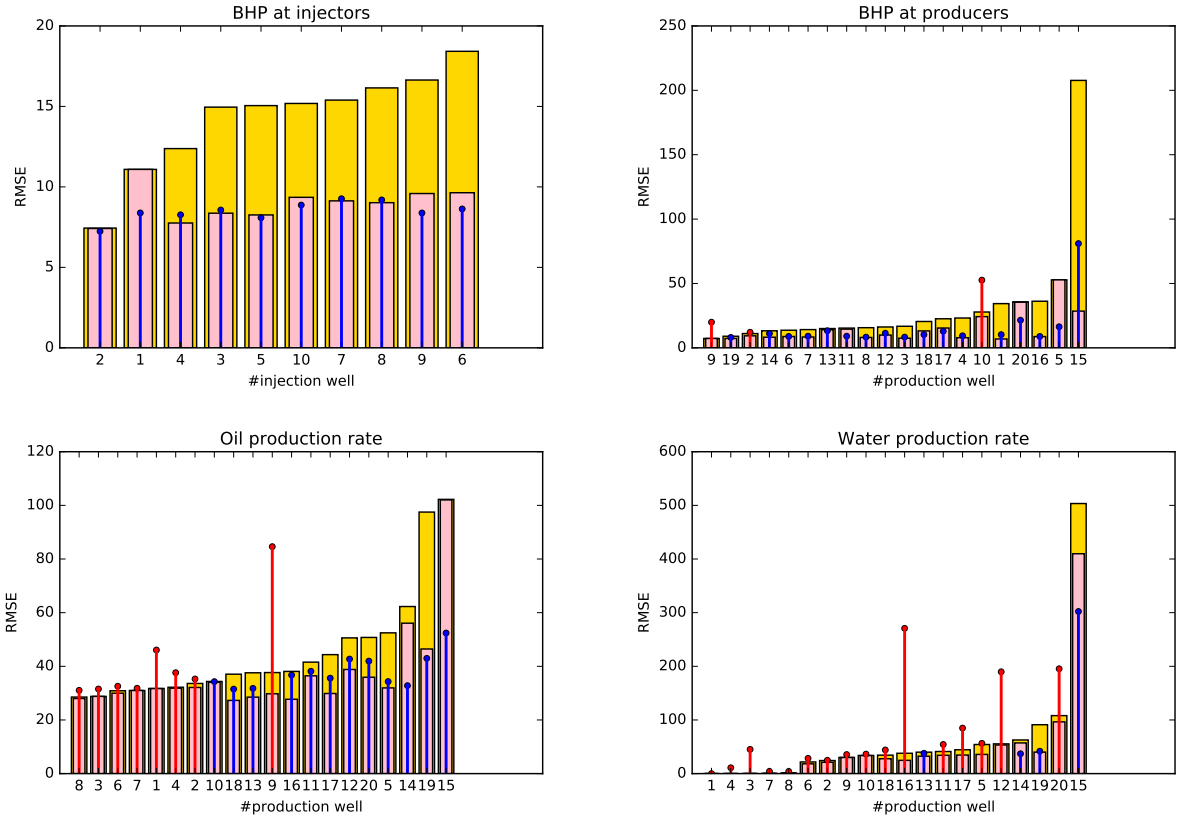


Figure 6.12: RMSEs of the best model (yellow bars) and of the best convex combination of models (pink bars) for each property, as well as the RMSEs of LASSO. The latter are depicted in blue whenever they are smaller than that of the best model for the considered property, in red otherwise.

As far as the selection is concerned, we actually study how often coefficients $w_{j,t}$ in the aggregation equations (6.3.1) are non-zero. It is well documented that LASSO outputs sparse vectors of weights $w_{j,t}$, i.e., that most of the coefficients picked are zero. Indeed, Figure 6.13 illustrates this fact while Figure 6.14 quantifies it. We read a high selection for the prediction of bottomhole pressure at injectors: only 4 models are active more than 20% of the time (when the averages are computed over time steps of the evaluation period 32 to 127, and over all properties of this family). The selection is even more extreme for oil production rate: no model is active more than 15% of the time. On the contrary, all models are active more than 40% of the time to predict the water production rate.

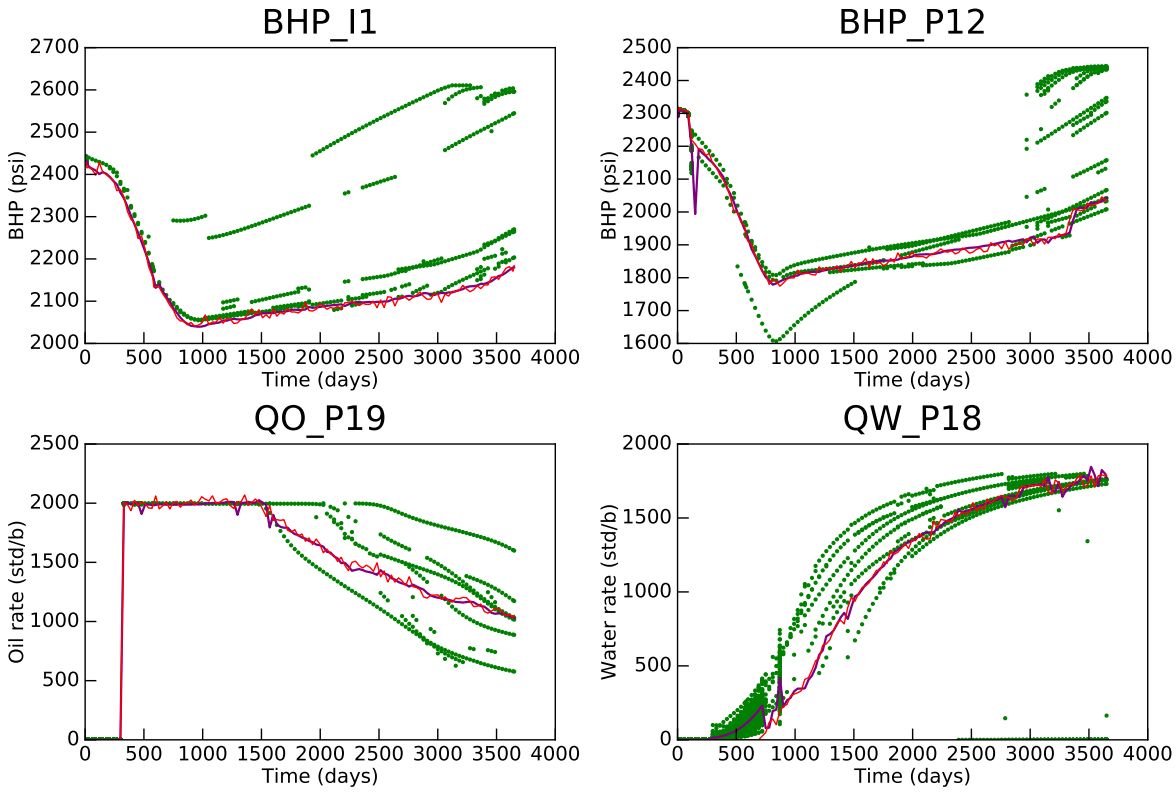


Figure 6.13: Illustration of the selection power of LASSO: same pictures and legend as in Figure 6.11, except that only forecasts of those models that are used in the aggregation are depicted; the forecasts associated with zero weights are omitted.

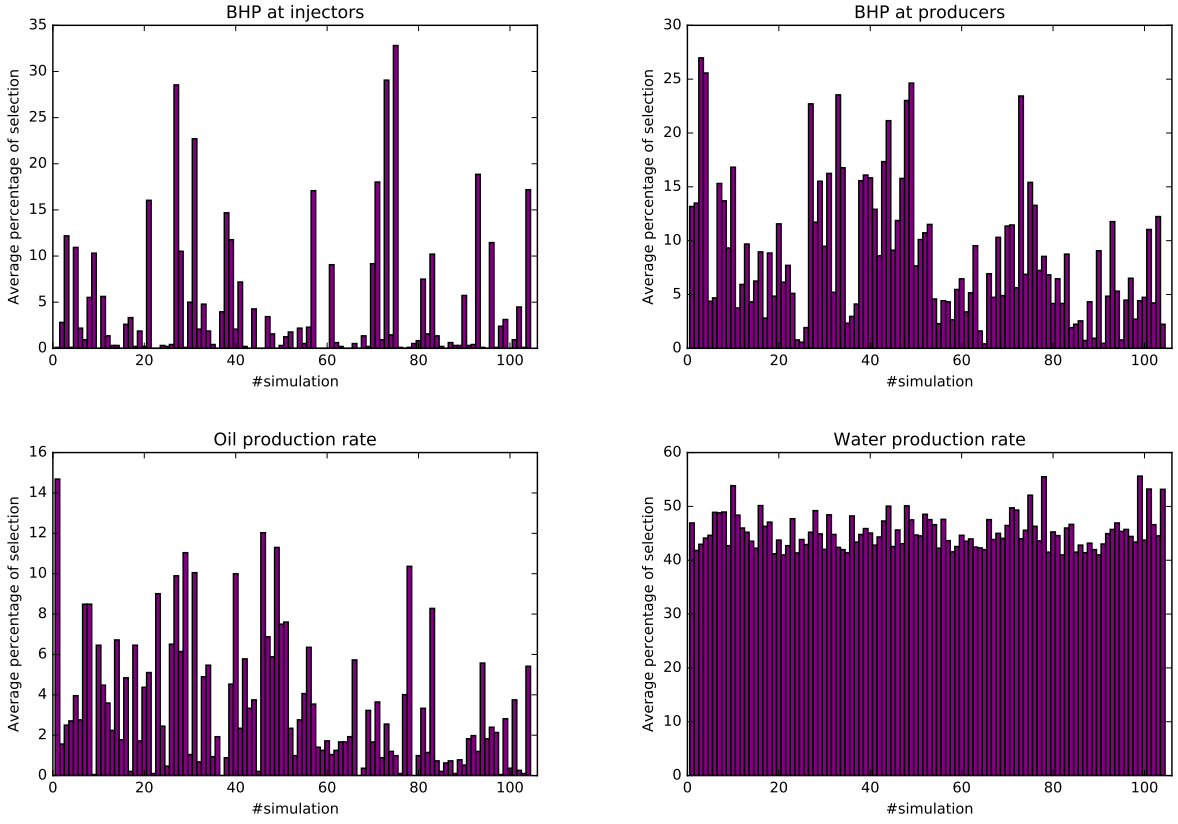


Figure 6.14: Illustration of the selection power of LASSO: average proportions (over time and properties) of models associated with non-zero coefficients in the aggregation. The models are indexed by integers between 1 and 104. The average proportions are computed by families of properties.

6.7. Appendix: Technical details for interval forecasts

These technical details are best described in Chapter 5 of this manuscript. They are of two sorts: a computational issue and some further specific descriptions on the general methodology explained in Section 6.5.

6.7.1. A computational issue to solve

First, a computational issue is to be discussed: getting a numerical value of the convex hull (6.3.3) is computationally challenging as S typically contains infinitely many scenarios. We could however provide a solution for two of the considered algorithms, namely Ridge and EWA, see Chapter 5.

For Ridge, we were able therein to determine a closed-form expression of the upper and lower bounds of the sets $\widehat{S}_{T+k}$ in (6.3.3).

As for EWA, we offer an efficient and recursive computation of a series of sets $\widehat{\mathcal{W}}'_{T+k}$ containing the target sets $\widehat{\mathcal{W}}_{T+k}$, from which it is then easy to compute intervals containing the target prediction intervals $\widehat{S}_{T+k}$. Indeed, it suffices to compute the maximum and the minimum of each $\widehat{S}_{T+k}$.

6.7.2. Some further specific descriptions on the methodology

Determining the set S of scenarios. We compute the maximal variations upwards M or downwards m of the observations on the learning part of the data set and of any single trajectory of model forecasts on the prediction part of the data set. We do so by considering variations averaged out over 10 consecutive steps. The maximal variation downwards m can be negative or positive, depending on the property considered; the same remark holds for the maximal variation upwards. The set S_{T+k} where the putative observations z_{T+k} lie is then equal to the interval $[y_T + k\,m,\, y_T + k\,M]$.

Correcting the interval forecasts for the noise level. We first study the learning part of the data set and estimate some upper bound $\sigma_{\max}$ on the noise level of the observations, as detailed below. Then, denoting by c_{T+k} the center of each interval forecast $\widehat{S}_{T+k}$, we replace

$$\widehat{S}_{T+k} \qquad \text{by} \qquad \max\Big\{\widehat{S}_{T+k},\ [c_{T+k} - \sigma_{\max},\, c_{T+k} + \sigma_{\max}]\Big\}\,,$$

where the maximum in the right-hand side has to be understood in terms of the inclusion $\subseteq$ operator.

Our estimate $\sigma_{\max}$ is formed as follows. We first determine, among the observations available, time steps corresponding to some local stability of the property studied; those steps t are the ones when the observation y_t is within 150 psi or 150 bbl/day (depending of the property) of all y_{t-r}, where r varies between -15 and $+15$. We denote by $\mathcal{S}$ the set of those time steps with local stability. Then, our estimate is

$$\sigma_{\max} = \max_{t\in\mathcal{S}} \left| y_t - \frac{1}{5}\sum_{r=t-2}^{t+2} z_r \right|.$$

Initial matching at T. The algorithm considered typically makes a prediction error when forecasting y_T by $\widehat{y}_T$ at the end of the learning part of the data set. To avoid that this error of $\Delta_T = \widehat{y}_T - y_T$ be carried over the whole prediction part of the data set (over all time steps $T + k$, where $k \geqslant 1$), we shift all interval forecasts $\widehat{S}_{T+k}$ by this initial error Δ_T (in the case of EWA) or by the average of $\Delta_{T-4}, \ldots, \Delta_T$ (in the case of Ridge).

Parameters λ and η. As indicated in Section 6.5, we could provide no totally satisfactory automatic choice of these parameters, unlike the procedure that we described in Section 6.3.3 for one-step-ahead prediction. However, two attempts are described, with their results, in the next section.

Selection of a subset of the models: best-performing ones. Finally, as far as Ridge is concerned (not EWA), we do not use all the models in the prediction part of the data set, but only the most reasonable ones: the ones whose root mean-square error on the learning part of the data set is smaller than 10 times the one of the best (also on this learning part) model. The forecasts of these models are printed in green while the forecasts of the models discarded due to this rule are in grey.

6.8. Supplementary material

In this section, we tackle the question of quantifying and evaluating the performance of forecast intervals algorithms. We therefore define a performance measure: the efficiency. We then introduce a benchmark.

In a third part, we present two ways of calibrating the regularization parameter for the Ridge forecast intervals, and compare their efficiency with the benchmark.

6.8.1. A performance measure for the forecast intervals

The aim of a forecast interval is to contain the future observation, so it seems natural to take into account the number (or the proportion) of observations lying inside the forecast intervals. But this count is not a satisfying measure on its own, since it can be trivially maximized, at least when there are known lower bound m and an upper bound M on the observations, by outputting the interval $[m, M]$. This interval is too wide: a good forecast interval should be narrow.

Therefore we introduce a measure that is based on this trade-off between width and accuracy of the interval forecasts: the efficiency. It is defined as the ratio of the number of observations lying within the forecast intervals, over the total sum of the width of the forecast intervals:

$$\text{Efficiency}\left(\widehat{S}_{T+1}, \widehat{S}_{T+2}, \dots\right) = \frac{\text{Card}\left\{k > 0 : y_{T+k} \in \widehat{S}_{T+k}\right\}}{\sum_k \left(\overline{\widehat{S}}_{T+k} - \underline{\widehat{S}}_{T+k}\right)}$$

where $\widehat{S}_{T+k} = \left[\underline{\widehat{S}}_{T+k}, \overline{\widehat{S}}_{T+k}\right]$ is the forecast interval at round $T+k$.

Obviously, the higher the efficiency, the better.

Let us emphasize the fact that this measure aims at comparing algorithms on the same data set, but its value for one precise algorithm, which depends for instance on the units, does not tell much about the quality of this algorithm. This measure should therefore be used in a relative way rather than an absolute way.

6.8.2. A possible benchmark

Classical benchmarks of sequential point aggregation (cf. (6.3.2)) can not be applied in the forecast intervals framework, since they forecast points and not intervals.

We suggest to use as benchmarks, the convex hulls of the forecasts of fixed subsets of experts. In order to keep computation time reasonable, we restrict ourselves to the subsets that contain only the best simulations on the learning sample. That is, we focus on the subsets of the K' simulations with the smallest RMSE on the learning sample, for any K' between 2 and K. Our benchmark $\text{eff}^\star$ will be the best efficiency among the efficiencies obtained using these $K-1$ subsets. It is formalized in Algorithm 16.

This benchmark will be used in the next section.

Algorithm 16 Computation of the forecast intervals benchmark

Input: The list of the indexes $n_1, \ldots, n_K$ of the simulations, sorted by increasing RMSE on the learning sample.
Initialization: $\text{eff}^\star = 0$.
for $K' = 2, \ldots, K$

1. **for** k=1,2,...
   (Subset K', interval k) Compute the following forecast interval:
$$\widehat{S}^{K'}_{T+k} = \text{Conv}\left(f_{n_1,T+k}, \ldots, f_{n_{K'},T+k}\right)$$
2. Compute the corresponding efficiency:
$$\text{Efficiency}(K') = \text{Efficiency}\left(\widehat{S}^{K'}_{T+1}, \widehat{S}^{K'}_{T+2}, \ldots\right)$$
3. Update the benchmark: $\text{eff}^\star = \max\left(\text{eff}^\star, \text{Efficiency}(K')\right)$

Output: $\text{eff}^\star$

6.8.3. Two tunings of the regularization parameter for the Ridge forecast intervals

We present, in this section, two ways of choosing the regularization parameter λ in the case of the Ridge forecast intervals.

An "objective-driven" tuning of the parameters

This tuning aims at getting a precise forecast interval with a chosen width W_{desired} (this will often make other forecast intervals have "reasonable" widths). We decided to work on the last forecast interval ($t = T_f$), for which we chose the width:

$$W_{\text{desired}} = \max\left(4\sigma, (\max_k f_{k,T_f} - \min_k f_{k,T_f})/10\right)$$

with σ the estimated noise level.

The tuning consists in three steps:

- 1. Choosing the desired width W_{desired} for a precise forecast interval.
- 2. Computing the forecast intervals corresponding to each regularization parameter belonging to a set of parameters (we used a grid of logarithmic-spaced parameters).
- 3. Keeping the parameter (and the corresponding forecast intervals) that leads to the width that is the closest to the desired width for the chosen forecast interval.

Actually, the widths of the forecast intervals tend to increase when the regularization parameter λ decreases, which can help speed up the computations.

A hybrid tuning

The tuning we present now is hybrid, in the sense that it mixes theoretical and empirical elements. It chooses the regularization parameter λ as the geometric mean of a fully empirical parameter λ_{emp} and a theory-based parameter λ_{the}: $\lambda = \sqrt{\lambda_{\text{emp}}\lambda_{\text{the}}}$.

λ_{emp} is the geometric mean of the parameters used by the "classical" version of the Ridge algorithm, for the last 10 instants of the learning sample (i.e., from $T-9$ to T).

$\lambda_{\text{the}} = 2FY\sqrt{T_f}K/B$ is an approximative minimizer of the bound of Theorem 2.7. F and Y are upper bounds respectively on the simulations and the observations; T_f is the total number of instants in the study; B is an upper bound on the Euclidean norm of the weight vectors. All these parameters are estimated using empirical data available at the instant of prediction (including simulations forecasts for instants posterior to T).

6.8.4. Results obtained with the two parameter tunings

Figure 6.15 shows the efficiencies (multiplied by 1000 for readability) of the Ridge forecast intervals, tuned with the two approaches of the previous section, compared to the efficiencies of the benchmark presented above. We recall that the higher the efficiency, the better. The efficiencies of the algorithms are drawn in blue when they are larger than the benchmark, in red when they are smaller. One can see improvable performance on the hardest properties (QW_P), but quite good performance for the other properties, at the level of the benchmark (QO_P) or even better (BHP_P, QO_P).

Figure 6.16 shows the interval forecasts obtained by the two tuning, on the four properties already studied. The forecast intervals obtained on these properties are rather accurate.

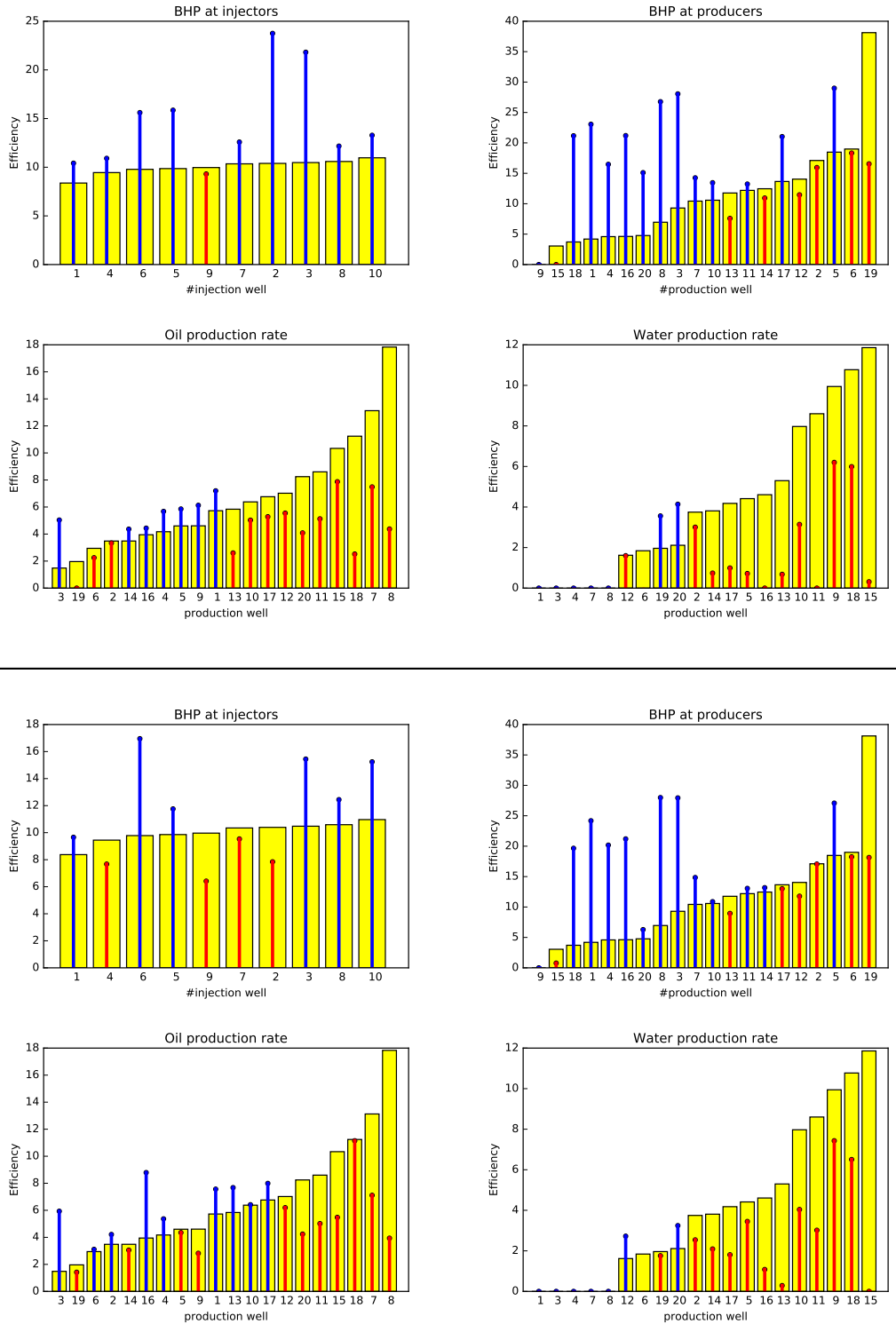


Figure 6.15: Efficiencies of the benchmark (yellow bars) for each property, as well as the efficiencies of the Ridge forecast intervals, using the "objective-driven" tuning (top graphs) and the " hybrid" tuning (bottom graphs). The Ridge efficiencies are depicted in blue whenever they are **larger** than that of the benchmark for the considered property, in red otherwise.

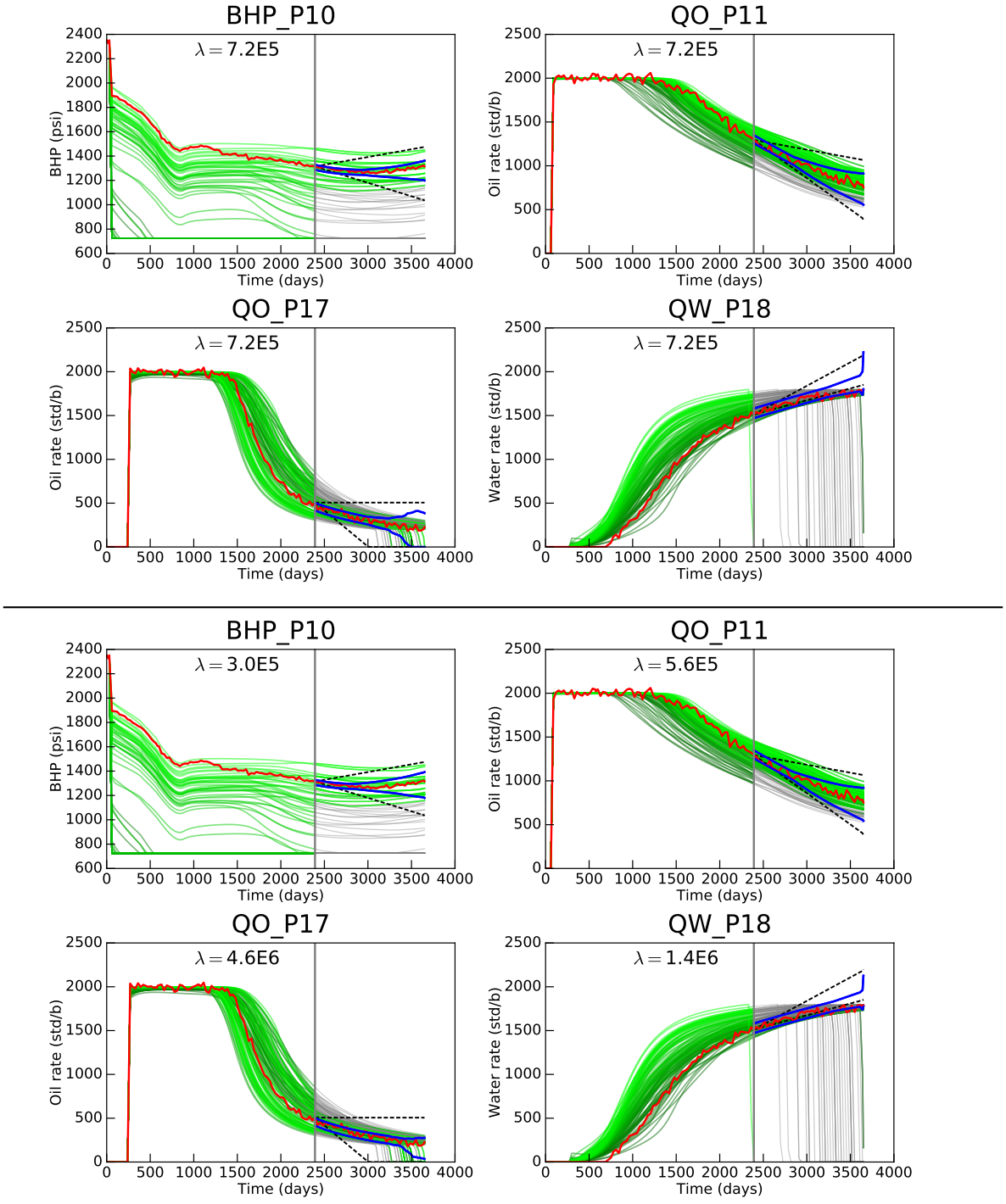


Figure 6.16: Simulations of the models (green lines — or grey lines —, depending on whether the simulations were selected for the interval forecasts), observations (red solid line —), set S of scenarios (upper and lower bounds given by black dotted lines - - -), and interval forecasts output by **Ridge** (upper and lower bounds given by blue solid lines —). Values of λ used are written on the graphs. The **top graphs** correspond to the **"objective-driven" tuning** of the regularization parameter, the **bottom graphs** to the **"hybrid" tuning**.

Bibliography

Sigurd I Aanonsen, Geir Nævdal, Dean S Oliver, Albert C Reynolds, and Brice Vallès. The ensemble Kalman filter in reservoir engineering –a review. SPE Journal, 14(03):393–412, 2009.

Dimitris Achlioptas. Database-friendly random projections. In Proceedings of the 20th ACM Symposium on Principles of Database Systems, 2001.

Dimitris Achlioptas and Frank McSherry. Fast computation of low-rank matrix approximations. Journal of the ACM, 54(2):9, 2007.

Hirotugu Akaike. A new look at the statistical model identification. IEEE transactions on automatic control, 19(6):716–723, 1974.

Christophe Amat, Tomasz Michalski, and Gilles Stoltz. Fundamentals and exchange rate forecastability with machine learning methods. 2016. Preprint; see http://halshs.archives-ouvertes.fr/halshs-01003914.

Sanjeev Arora, Elad Hazan, and Satyen Kale. A fast random sampling algorithm for sparsifying matrices. In Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, pages 272–279. Springer, 2006.

Jean-Yves Audibert. Fast learning rates in statistical inference through aggregation. The Annals of Statistics, 37(4):1591–1646, 2009.

P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. Journal of Computer and System Sciences, 64:48–75, 2002.

Katy S Azoury and Manfred K Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. Machine Learning, 43(3):211–246, 2001.

Jushan Bai and Serena Ng. Forecasting economic time series using targeted predictors. Journal of Econometrics, 146(2):304–317, 2008.

Shamas ul Islam Bajwa, Edward Chung, and Masao Kuwahara. Performance evaluation of an adaptive travel time prediction model. In Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems, 2005.

Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. Constructive Approximation, 28(3):253–263, 2008.

Andrew Barron, Lucien Birgé, and Pascal Massart. Risk bounds for model selection via penalization. Probability Theory and Related Fields, 113(3):301–413, 1999.

Peter Bartlett. Online prediction, 2011. Lectures at IHP in May 2011. Slides available at http://www.stat.berkeley.edu/~bartlett/talks/ihp-may-2011.pdf.

Peter L. Bartlett and Shahar Mendelson. Empirical minimization. Probability Theory and Related Fields, 135 (3):311–334, 2006.

Pierre Bellec. Localized Gaussian width of M-convex hulls with applications to Lasso and convex aggregation. Technical report, Rutgers University, 2017.

Pierre Bellec, Guillaume Lecué, and Alexandre Tsybakov. Slope meets Lasso: improved oracle bounds and optimality. Technical report, CNRS, Université Paris-Saclay, CREST, 2016.

Marie Bessec. Étalonnages du taux de croissance du PIB français sur la base des enquêtes de conjoncture. Economie & prévision, (2):77–99, 2010.

Peter J. Bickel, Ya'acov Ritov, and Alexandre B. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. The Annals of Statistics, 37(4):1705–1732, 2009.

Lucien Birgé and Pascal Massart. Gaussian model selection. Journal of the European Mathematical Society, 3(3):203–268, 2001.

Avrim Blum. Empirical support for Winnow and Weighted-Majority algorithms: Results on a calendar scheduling domain. Machine Learning, 26:5–23, 1997.

Christos Boutsidis, Michael W Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In Proceedings of the 20th annual ACM-SIAM Symposium on Discrete Algorithms, pages 968–977, 2009.

Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal column-based matrix reconstruction. SIAM Journal on Computing, 43(2):687–717, 2014.

George E.P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. Time Series Analysis: Forecasting and Control. John Wiley & Sons, 2015.

Stephen Boyd and Lieven Vandenberghe. Convex optimization. Cambridge University Press, 2004.

Peter Bühlmann and Sara van de Geer. Statistics for high-dimensional data. Springer Series in Statistics. Springer, Heidelberg, 2011. Methods, theory and applications.

Emmanuel Candes and Terence Tao. The Dantzig selector: Statistical estimation when p is much larger than n. The Annals of Statistics, pages 2313–2351, 2007.

Emmanuel J Candès and Yaniv Plan. Near-ideal model selection by ℓ 1 minimization. The Annals of Statistics, 37(5A):2145–2177, 2009.

Emmanuel J. Candes and Terence Tao. Decoding by linear programming. Institute of Electrical and Electronics Engineers. Transactions on Information Theory, 51(12):4203–4215, 2005.

Emmanuel J. Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. Institute of Electrical and Electronics Engineers. Transactions on Information Theory, 56(5):2053–2080, 2010.

Emmanuel J. Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. Institute of Electrical and Electronics Engineers. Transactions on Information Theory, 52(2):489–509, 2006.

N. Cesa-Bianchi, Y. Freund, D. Haussler, D.P. Helmbold, R. Schapire, and M. Warmuth. How to use expert advice. Journal of the ACM, 44(3):427–485, 1997.

Nicolò Cesa-Bianchi. Analysis of two gradient-based algorithms for on-line regression. Journal of Computer and System Sciences, 59(3):392–411, 1999.

Nicolò Cesa-Bianchi and Gábor Lugosi. Prediction, Learning, and Games. Cambridge University Press, 2006.

Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. Machine Learning, 66(2-3):321–352, 2007.

Djalil Chafaï, Olivier Guédon, Guillaume Lecué, and Alain Pajor. Interactions between compressed sensing random matrices and high dimensional geometry, volume 37 of Panoramas et Synthèses. Société Mathématique de France, Paris, 2012.

Ta-Chien Chan, Tsuey-Hwa Hu, and Jing-Shiang Hwang. Daily forecast of dengue fever incidents for urban villages in a city. International Journal of Health Geographics, 14(1):9, 2015.

Chris Chatfield. Prediction intervals for time-series forecasting. In Principles of forecasting, pages 475–494. Springer, 2001.

Sourav Chatterjee. A new perspective on least squares under convex constraint. The Annals of Statistics, 42 (6):2340–2381, 2014.

Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. SIAM review, 43(1):129–159, 2001.

Peter F. Christoffersen. Evaluating interval forecasts. International Economic Review, pages 841–862, 1998.

Mark A Davenport and Michael B Wakin. Analysis of orthogonal matching pursuit using the restricted isometry property. IEEE Transactions on Information Theory, 56(9):4395–4401, 2010.

Steven De Rooij, Tim Van Erven, Peter D Grünwald, and Wouter M Koolen. Follow the leader if you can, hedge if you must. The Journal of Machine Learning Research, 15(1):1281–1316, 2014.

Marie Devaine, Pierre Gaillard, Yannig Goude, and Gilles Stoltz. Forecasting the electricity consumption by aggregation of specialized experts; application to Slovakian and French country-wide (half-)hourly predictions. Machine Learning, 90(2):231–260, 2013.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(Jul):2121–2159, 2011.

B. Efron, I. Johnstone, T. Hastie, and R. Tibshirani. Least angle regression. Annals of Statistics, 32(2): 407–499, 2004.

Simon Foucart and Holger Rauhut. A mathematical introduction to compressive sensing. Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, New York, 2013.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119–139, 1997.

Yoav Freund, Robert E Schapire, Yoram Singer, and Manfred K Warmuth. Using and combining predictors that specialize. In Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, pages 334–343. ACM, 1997.

Pierre Gaillard. Contributions à l'agrégation séquentielle robuste d'experts : travaux sur l'erreur d'approximation et la prévision en loi. Applications à la prévision pour les marchés de l'énergie. PhD thesis, Université Paris-Sud 11, 2015.

Pierre Gaillard and Yannig Goude. Forecasting the electricity consumption by aggregating experts; how to design a good set of experts. In Modeling and Stochastic Learning for Forecasting in High Dimension, Lecture Notes in Statistics. Springer, 2015.

Pierre Gaillard, Gilles Stoltz, and Tim Van Erven. A second-order bound with excess losses. In Proceedings of COLT'14, pages 176–196, 2014.

Francis Galton. Natural inheritance. Macmillan and Company, 1894.

Rahul Garg and Rohit Khandekar. Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 337–344. ACM, 2009.

Andrey Y Garnaev and Efim D Gluskin. The widths of a Euclidean ball. Doklady Akademii Nauk SSSR, 277 (5):1048–1052, 1984.

S. Gerchinovitz. Prediction of individual sequences and prediction in the statistical framework: some links around sparse regression and aggregation techniques. PhD thesis, Université Paris-Sud, Orsay, 2011.

Mina Ghashami, Edo Liberty, Jeff M Phillips, and David P Woodruff. Frequent directions: Simple and deterministic matrix sketching. SIAM Journal on Computing, 45(5):1762–1792, 2016.

Christophe Giraud. Introduction to high-dimensional statistics, volume 138. CRC Press, 2014.

Eyal Gofer, Nicolò Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Regret minimization for branching experts. In Proceedings of the 26–th Conference on Learning Theory (COLT '13), pages 618–638, 2013.

Suleyman Gokcan. Forecasting volatility of emerging stock markets: linear versus non-linear GARCH models. Journal of Forecasting, 19(6):499–504, 2000.

Maya R Gupta, Samy Bengio, and Jason Weston. Training highly multiclass classifiers. Journal of Machine Learning Research, 15:1461–1492, 2014.

Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. Theory of computing, 8(1):321–350, 2012.

Elad Hazan. Introduction to online convex optimization. Foundations and Trends in Optimization, 2(3-4): 157–325, 2016.

Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. Journal of Machine Learning Research, 15(1):2489–2512, 2014.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. Machine Learning, 69(2):169–192, 2007.

Mark Herbster and Manfred K Warmuth. Tracking the best expert. Machine learning, 32(2):151–178, 1998.

A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12:55–67, 1970.

Arthur E. Hoerl. Application of ridge analysis to regression problems. Chemical Engineering Progress, 58(3): 54–59, 1962.

Vladimir Koltchinskii. Local Rademacher complexities and oracle inequalities in risk minimization. The Annals of Statistics, 34(6):2593–2656, 2006.

Vladimir Koltchinskii. Oracle inequalities in empirical risk minimization and sparse recovery problems, volume 2033 of Lecture Notes in Mathematics. Springer, Heidelberg, 2011. Lectures from the 38th Probability Summer School held in Saint-Flour, 2008, École d'Été de Probabilités de Saint-Flour.

Wouter M. Koolen and Tim Van Erven. Second-order quantile methods for experts and combinatorial games. In Proceedings of COLT'15, volume 40, pages 1155–1175, 2015.

Markku Lanne and Pentti Saikkonen. Noncausal vector autoregression. Econometric Theory, 29(03):447–481, 2013.

Guillaume Lecué. Interplay between concentration, complexity and geometry in learning theory with applications to high dimensional data analysis. Habilitation à diriger des recherches. 2011.

Guillaume Lecué and Shahar Mendelson. Learning subgaussian classes: Upper and minimax bounds. arXiv preprint arXiv:1305.4825, 2013.

Guillaume Lecué and Shahar Mendelson. Regularization and the small-ball method I: sparse recovery. arXiv preprint arXiv:1601.05584, 2016.

Guillaume Lecué and Shahar Mendelson. Sparse recovery under weak moment assumptions. Journal of the European Mathematical Society (JEMS), 19(3):881–904, 2017.

Michel Ledoux. The concentration of measure phenomenon, volume 89 of Mathematical Surveys and Monographs. American Mathematical Society, Providence, RI, 2001.

Michel Ledoux and Michel Talagrand. Probability in Banach Spaces: isoperimetry and processes. Springer, 2013.

Edo Liberty. Simple and deterministic matrix sketching. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 581–588, 2013.

Nick Littlestone. From on-line to batch learning. In Proceedings of the second annual workshop on Computational Learning Theory, pages 269–284. Morgan Kaufmann Publishers, 1989.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. Information and Computation, 108(2):212–261, 1994.

Yiming Lu, Yang Zhou, Wubin Qu, Minghua Deng, and Chenggang Zhang. A Lasso regression model for the construction of microRNA-target regulatory networks. Bioinformatics, 27(17):2406–2413, 2011.

Haipeng Luo, Alekh Agarwal, Nicolò Cesa-Bianchi, and John Langford. Efficient second order online learning by sketching. In Advances in Neural Information Processing Systems (NIPS), pages 902–910, 2016.

Stéphane Mallat. A wavelet tour of signal processing. Elsevier/Academic Press, Amsterdam, third edition, 2009. The sparse way, With contributions from Gabriel Peyré.

Colin L Mallows. Some comments on C_p. Technometrics, 15(4):661–675, 1973.

Pascal Massart. Concentration inequalities and model selection, volume 1896 of Lecture Notes in Mathematics. Springer, Berlin, 2007. Lectures from the 33rd Summer School on Probability Theory held in Saint-Flour, July 6–23, 2003, With a foreword by Jean Picard.

Boris Mauricette, Vivien Mallet, and Gilles Stoltz. Ozone ensemble forecast with machine learning algorithms. Journal of Geophysical Research: Atmospheres, 114(D5), 2009.

Shahar Mendelson. Upper bounds on product and multiplier empirical processes. Stochastic Processes and their Applications, 126(12):3652–3680, 2016.

Isobel Milns, Colin M. Beale, and V. Anne Smith. Revealing ecological networks using Bayesian network inference algorithms. Ecology, 91(7):1892–1899, 2010.

Jayadev Misra and David Gries. Finding repeated elements. Science of computer programming, 2(2):143–152, 1982.

Balas Kausik Natarajan. Sparse approximate solutions to linear systems. SIAM journal on computing, 24(2): 227–234, 1995.

Arkadii Nemirovski. Lectures on probability theory and statistics, volume 1738 of Lecture Notes in Mathematics. Springer-Verlag, Berlin, 2000. Lectures from the 28th Summer School on Probability Theory held in Saint-Flour, August 17–September 3, 1998, Edited by Pierre Bernard.

Dean S Oliver and Yan Chen. Recent progress on reservoir history matching: a review. Computational Geosciences, 15(1):185–221, 2011.

Karl Pearson. Mathematical contributions to the theory of evolution. III. Regression, heredity, and panmixia. Philosophical Transactions of the Royal Society of London. Series A, 187:253–318, 1896.

Lies Peters, Rob Arts, Geert Brouwer, Cees Geel, Stan Cullick, Rolf J Lorentzen, Yan Chen, Neil Dunlop, Femke C Vossepoel, Rong Xu, Pallav Sarma, Ahmed H H Alhuthali, and Albert Reynolds. Results of the Brugge benchmark study for flooding optimization and history matching. SPE Reservoir Evaluation & Engineering, 13(03):391–405, 2010.

Allan Pinkus. n-widths in approximation theory, volume 7 of Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]. Springer-Verlag, Berlin, 1985.

Nalini Ravishanker, L. Shiao-Yen Wu, and Joseph Glaz. Multiple prediction intervals for time series: comparison of simultaneous and marginal intervals. Journal of Forecasting, 10(5):445–463, 1991.

Philippe Rigollet and Alexandre Tsybakov. Exponential screening and optimal rates of sparse estimation. The Annals of Statistics, 39(2):731–771, 2011.

Saskia Rinke and Philipp Sibbertsen. Information criteria for nonlinear time series models. Studies in Nonlinear Dynamics and Econometrics, 20(3):325–341, 2016.

Angelika Rohde and Alexandre B. Tsybakov. Estimation of high-dimensional low-rank matrices. The Annals of Statistics, 39(2):887–930, 2011.

Mattia Saccoccio, Ting Hei Wan, Chi Chen, and Francesco Ciucci. Optimal regularization in distribution of relaxation times applied to electrochemical impedance spectroscopy: Ridge and Lasso regression methods –a theoretical and experimental study. Electrochimica Acta, 147:470–482, 2014.

Aaditya Satija and Jef Caers. Direct forecasting of subsurface flow response from non-linear dynamic data by linear least-squares in canonical functional principal component space. Advances in Water Resources, 77: 69–81, 2015.

Addy Satija, Céline Scheidt, Lewis Li, and Jef Caers. Direct forecasting of reservoir performance using production data without history matching. Computational Geosciences, 21(2):315–333, 2017.

Adrien Saumard. Optimal upper and lower bounds for the true and empirical excess risks in heteroscedastic least-squares regression. Electronic Journal of Statistics, 6(1-2):579–655, 2012.

Adrien Saumard. A concentration inequality for the excess risk in least-squares regression with random design and heteroscedastic noise. arXiv preprint arXiv:1702.05063, 2017.

Céline Scheidt, Philippe Renard, and Jef Caers. Prediction-focused subsurface modeling: investigating the need for accuracy in flow-based inverse modeling. Mathematical Geosciences, 47(2):173–191, 2015.

Gideon Schwarz. Estimating the dimension of a model. The Annals of Statistics, 6(2):461–464, 1978.

Ralph D. Snyder, J. Keith Ord, and Anne B. Koehler. Prediction intervals for ARIMA models. Journal of Business and Economic Statistics, 19(2):217–225, 2001.

Jeffrey M Stanton. Galton, Pearson, and the peas: A brief history of linear regression for statistics instructors. Journal of Statistics Education, 9(3), 2001.

Charles Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In Proceedings of the Third Berkeley symposium on mathematical statistics and probability, volume 1, pages 197–206, 1956.

Stephen M Stigler. The history of statistics: The measurement of uncertainty before 1900. Harvard University Press, 1986.

Wenyue Sun and Louis J Durlofsky. A new data-space inversion procedure for efficient uncertainty quantification in subsurface flow problems. Mathematical Geosciences, pages 1–37, 2017.

Michel Talagrand. Upper and lower bounds for stochastic processes, volume 60 of Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge. [Results in Mathematics and Related Areas 3rd Series. A Series of Modern Surveys in Mathematics]. Springer, Heidelberg, 2014. Modern methods and classical problems.

Albert Tarantola. Inverse problem theory: Method for data fitting and model parameter estimation. Elsevier, 613, 1987.

Robert Tibshirani. Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society. Series B. Methodological, pages 267–288, 1996.

AN Tikhonov and Vasili Ya Arsenin. Methods for solving ill-posed problems. John Wiley and Sons, Inc, 1977.

Alexandre B. Tsybakov. Optimal rates of aggregation. In Proceedings of COLT'03, Lecture Notes in Artificial Intelligence, pages 303–313. 2003.

Alexandre B. Tsybakov. Introduction to nonparametric estimation. Springer Series in Statistics. Springer, New York, 2009. Revised and extended from the 2004 French original, Translated by Vladimir Zaiats.

Sara A. van de Geer. Applications of empirical process theory, volume 6 of Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 2000.

Sara A. van de Geer. The deterministic Lasso. Technical report, ETH Zürich, 2007. http://www.stat.math.ethz.ch/ geer/lasso.pdf.

Sara A Van De Geer and Peter Bühlmann. On the conditions used to prove oracle results for the Lasso. Electronic Journal of Statistics, 3:1360–1392, 2009.

Aad W. van der Vaart and Jon A. Wellner. Weak convergence and empirical processes. Springer Series in Statistics. Springer-Verlag, New York, 1996. With applications to statistics.

Tim van Erven and Wouter M Koolen. MetaGrad: Multiple Learning Rates in Online Learning. In Advances in Neural Information Processing Systems, pages 3666–3674, 2016.

Vladimir N. Vapnik. Statistical learning theory. Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, Inc., New York, 1998. A Wiley-Interscience Publication.

R Vernet. A model to forecast data centre infrastructure costs. In Journal of Physics: Conference Series, volume 664, page 052040. IOP Publishing, 2015.

Nicolas Verzelen. Minimax risks for sparse regressions: ultra-high dimensional phenomenons. Electronic Journal of Statistics, 6:38–90, 2012.

Hai X Vo and Louis J Durlofsky. Regularized kernel pca for the efficient parameterization of complex geological models. Journal of Computational Physics, 322:859–881, 2016.

V. Vovk. Aggregating strategies. In Proceedings of the Third Annual Workshop on Computational Learning Theory (COLT), pages 372–383, 1990.

Volodya Vovk. Competitive on-line statistics. International Statistical Review, 69(2):213–248, 2001.

Can Wan, Zhao Xu, Pierre Pinson, Zhao Yang Dong, and Kit Po Wong. Optimal prediction intervals of wind power generation. IEEE Transactions on Power Systems, 29(3):1166–1174, 2014.

Rafał Weron. Estimating long-range dependence: finite sample properties and confidence intervals. Physica A: Statistical Mechanics and its Applications, 312(1-2):285–299, 2002.

David P Woodruff. Sketching as a tool for numerical linear algebra. Foundations and Trends in Theoretical Computer Science, 10(1–2):1–157, 2014.

Audrey W Zhu and Halton Pi. A method for improving the accuracy of weather forecasts based on a comprehensive statistical analysis of historical data for the contiguous United States. Journal of Climatology & Weather Forecasting, 2014.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In Proceedings of the 20th International Conference on Machine Learning (ICML-03), pages 928–936, 2003.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B. Statistical Methodology, 67(2):301–320, 2005.

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Titre : Régression linéaire et apprentissage : contributions aux méthodes de régularisation et d'agrégation

Mots Clefs : Apprentissage, régression linéaire, régularisation, agrégation, processus empiriques, optimisation convexe séquentielle

Résumé : Cette thèse aborde le sujet de la régression linéaire dans différents cadres, liés notamment à l'apprentissage. Les deux premiers chapitres présentent le contexte des travaux, leurs apports et les outils mathématiques utilisés. Le troisième chapitre est consacré à la construction d'une fonction de régularisation optimale, permettant par exemple d'améliorer sur le plan théorique la régularisation de l'estimateur LASSO. Le quatrième chapitre présente, dans le domaine de l'optimisation convexe séquentielle, des accélérations d'un algorithme récent et prometteur, MetaGrad, et une conversion d'un cadre dit "séquentiel déterministe" vers un cadre dit "batch stochastique" pour cet algorithme. Le cinquième chapitre s'intéresse à des prévisions successives par intervalles, fondées sur l'agrégation de prédicteurs, sans retour d'expérience intermédiaire ni modélisation stochastique. Enfin, le sixième chapitre applique à un jeu de données pétrolières plusieurs méthodes d'agrégation, aboutissant à des prévisions ponctuelles court-terme et des intervalles de prévision long-terme.

Title: Linear regression and learning: contributions to regularization and aggregation methods

Keys words: Learning, linear regression, regularization, aggregation, empirical processes, online convex optimization

Abstract: This thesis tackles the topic of linear regression, within several frameworks, mainly linked to statistical learning. The first and second chapters present the context, the results and the mathematical tools of the manuscript. In the third chapter, we provide a way of building an optimal regularization function, improving for instance, in a theoretical way, the LASSO estimator. The fourth chapter presents, in the field of online convex optimization, speed-ups for a recent and promising algorithm, MetaGrad, and shows how to transfer its guarantees from a so-called "online deterministic setting" to a "stochastic batch setting". In the fifth chapter, we introduce a new method to forecast successive intervals by aggregating predictors, without intermediate feedback nor stochastic modeling. The sixth chapter applies several aggregation methods to an oil production dataset, forecasting short-term precise values and long-term intervals.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France